STUDIES IN LOGIC
AND
PRACTICAL REASONING

VOLUME 3

D.M GABBAY / P. GARDENFORS / J. SIEKMANN / J. VAN BENTHEM / M. VARDI / J. WOODS
EDITORS

Handbook of Modal Logic

EDITED BY
P. BLACKBURN
J. VAN BENTHEM
F. WOLTER

HANDBOOK OF MODAL LOGIC

STUDIES IN LOGIC

AND

PRACTICAL REASONING

VOLUME 3

Editors:

D.M. GABBAY, *London*
P. GÄRDENFORS, *Lund*
J. SIEKMANN, *Saarbrücken*
J. VAN BENTHEM, *Amsterdam & Stanford*
M. VARDI, *Houston*
J. WOODS, *Vancouver & London*

A list of titles in this series appears at the end of this Volume

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON • NEW YORK • OXFORD
PARIS • SAN DIEGO • SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

HANDBOOK OF MODAL LOGIC

Edited by

Patrick BLACKBURN
INRIA Lorraine, France

Johan VAN BENTHEM
University of Amsterdam, The Netherlands
and
Stanford University, U.S.A.

Frank WOLTER
University of Liverpool, U.K.

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON • NEW YORK • OXFORD
PARIS • SAN DIEGO • SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Elsevier
Radarweg 29, PO Box 211, 1000 AE Amsterdam, The Netherlands
The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, UK

First edition 2007

Copyright © 2007 Elsevier B.V. All rights reserved

No part of this publication may be reproduced, stored in a retrieval system
or transmitted in any form or by any means electronic, mechanical, photocopying,
recording or otherwise without the prior written permission of the publisher

Permissions may be sought directly from Elsevier's Science & Technology Rights
Department in Oxford, UK: phone (+44) (0) 1865 843830; fax (+44) (0) 1865 853333;
email: permissions@elsevier.com. Alternatively you can submit your request online by
visiting the Elsevier web site at http://elsevier.com/locate/permissions, and selecting
Obtaining permission to use Elsevier material

Notice
No responsibility is assumed by the publisher for any injury and/or damage to persons
or property as a matter of products liability, negligence or otherwise, or from any use
or operation of any methods, products, instructions or ideas contained in the material
herein. Because of rapid advances in the medical sciences, in particular, independent
verification of diagnoses and drug dosages should be made

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN-13: 978-0-444-51690-9
ISBN-10: 0-444-51690-5
ISSN: 1570-2464

For information on all Elsevier publications
visit our website at books.elsevier.com

Printed and bound in The Netherlands

07 08 09 10 11 10 9 8 7 6 5 4 3 2 1

CONTENTS

List of Contributors vii

Preface xi
P. Blackburn, J. van Benthem and F. Wolter

Part 1. Basic Theory

1 Modal Logic: A Semantic Perspective 1
P. Blackburn and J. van Benthem

2 Modal Proof Theory 85
M. Fitting

3 Complexity of Modal Logic 139
M. Marx

4 Computational Modal Logic 181
I. Horrocks, U. Hustadt, U. Sattler, and R. Schmidt

Part 2. Advanced Theory

5 Model Theory of Modal Logic 249
V. Goranko and M. Otto

6 Algebras and Co-algebras 331
Y. Venema

7 Modal Decision Problems 427
F. Wolter and M. Zakharyaschev

8 Modal Consequence Relations 491
M. Kracht

Part 3. Variations and Extensions

9 First-order Modal Logic 549
T. Braüner and S. Ghilardi

10 Higher-order Modal Logic 621
R. Muskens

11 Temporal Logic 655
I. Hodkinson and M. Reynolds

12 Modal μ-calculi 721
J. Bradfield and C. Stirling

13 Description Logic 757
F. Baader and C. Lutz

14 Hybrid Logics 821
C. Areces and B. ten Cate

15 Combining Modal Logics 869
A. Kurucz

Part 4. Applications

16 Modal Logic in Mathematics 927
S. Artemov

17 Automata-theoretic Techniques for Temporal Reasoning 971
M. Vardi

18 Intelligent Agents and Common-sense Reasoning 991
J.-J. Meyer and F. Veltman

19 Applications of Modal Logic in Linguistics 1031
L. Moss and H.-J. Tiede

20 Modal Logic for Games and Information 1077
W. van der Hoek and M. Pauly

21 Modal Logic and Philosophy 1149
S. Lindström and K. Segerberg

Index 1215

CONTRIBUTORS

Carlos Areces
INRIA Lorraine, 615 rue du Jardin Botanique, 54602 Villers lès Nancy Cedex, France. carlos.areces@loria.fr

Sergei Artemov
Computer Science, Graduate Center CUNY, 365 Fifth Avenue, New York, NY 10016, USA. sartemov@gc.cuny.edu

Franz Baader
TU Dresden, Department of Computer Science, Institute for Theoretical Computer Science, 01062 Dresden, Germany. baader@tcs.inf.tu-dresden.de

Johan van Benthem
Institute for Logic, Language & Computation, (ILLC), University of Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands; and Department of Philosophy, Stanford University, Stanford, CA 94305, USA. johan@science.uva.nl

Patrick Blackburn
INRIA Lorraine, 615 rue du Jardin Botanique, 54602 Villers lès Nancy Cedex, France. patrick.blackburn@loria.fr

Julian Bradfield
Computer Science, King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, UK. jcb@inf.ed.ac.uk

Torben Braüner
Department of Computer Science, Roskilde University, PO Box 260, DK-4000 Roskilde, Denmark. torben@ruc.dk

Balder ten Cate
ISLA - Informatics Institute, University of Amsterdam, Kruislaan 403, 1098SJ, Amsterdam, The Netherlands. balder.tencate@uva.nl

Melvin Fitting
Department of Mathematics and Computer Science, Lehman College, 250 Bedford Park Blvd. West, Bronx, NY 10468-1589, USA. melvin.fitting@lehman.cuny.edu

Silvio Ghilardi
Department of Computer Science, Università degli Studi di Milano, via Comelico 39, 20135 Milano, Italy. ghilardi@dsi.unimi.it

Valentin Goranko
School of Mathematics, University of the Witwatersrand, Private Bag 3, WITS 2050, Johannesburg, South Africa. goranko@maths.wits.ac.za

Ian Hodkinson
Department of Computing, Imperial College London, South Kensington Campus, London, SW7 2AZ, UK. imh@doc.ic.ac.uk

Wiebe van der Hoek
Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK. wiebe@csc.liv.ac.uk

Ian Horrocks
School of Computer Science, University of Manchester, Oxford Road, Manchester, M13 9PL, UK. horrocks@cs.man.ac.uk

Ullrich Hustadt
Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK. u.hustadt@csc.liv.ac.uk

Marcus Kracht
Department of Linguistics, UCLA, 405 Hilgard Avenue, 3125 Campbell Hall, Los Angeles, CA 90095-1543, USA. kracht@humnet.ucla.edu

Agi Kurucz
Department of Computer Science, King's College London, Strand, London WC2R 2LS, UK. agi.kurucz@kcl.ac.uk

Sten Lindström
Umeå universitet, Institutionen för filosofi och lingvistik, S 901 87 Umeå, Sweden. sten.lindstrom@philos.umu.se

Carsten Lutz
TU Dresden, Department of Computer Science, Institute for Theoretical Computer Science, 01062 Dresden, Germany. lutz@tcs.inf.tu-dresden.de

Maarten Marx
ISLA, Informatics Institute, Universiteit van Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands. marx@science.uva.nl

John-Jules Meyer
Institute of Information and Computer Science, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, The Netherlands. jj@cs.uu.nl

Lawrence S. Moss
Department of Mathematics, Indiana University, Bloomington, IN 47405-7106, USA. lsm@cs.indiana.edu

Reinhard Muskens
Faculty of Philosophy, Tilburg University, PO Box 90153, NL 5000 LE Tilburg, The Netherlands. r.a.muskens@uvt.nl

Martin Otto
Mathematics Department, Technische Universität Darmstadt, Schlossgartenstr. 7, D-64289 Darmstadt, Germany. otto@mathematik.tu-darmstadt.de

Marc Pauly
Department of Philosophy, Stanford University, Stanford, CA 94305-2155, USA. marc.pauly@stanford.edu

Mark Reynolds
School of Computer Science & Software Engineering, Faculty of Engineering, Computing and Mathematics, The University of Western Australia, M002, 35 Stirling Highway, Crawley, WA 6009, Australia. mark@csse.uwa.edu.au

Ulrike Sattler
School of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, UK. sattler@cs.man.ac.uk

Krister Segerberg
Filosofiska institutionen, Uppsala Universitet, Box 627, 751 26 Uppsala, Sweden. krister.segerberg@filosofi.uu.se

Renate Schmidt
School of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, UK. renate.schmidt@manchester.ac.uk

Colin Stirling
Computer Science, King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, UK.
cps@inf.ed.ac.uk

Hans-Jörg Tiede
Illinois Wesleyan University, Department of Mathematics and Computer Science, PO Box 2900, Bloomington, IL, 61702-2900, USA. htiede@iwu.edu

Moshe Y. Vardi
Department of Computer Science, Rice University, 6100 S. Main Street, Houston, TX 77005-1892, USA. vardi@cs.rice.edu

Frank Veltman
Institute for Logic, Language and Computation, University of Amsterdam, Nieuwe Doelenstraat 15, 1012 CP Amsterdam, The Netherlands. f.veltman@uva.nl

Yde Venema
Institute for Logic, Language and Computation, University of Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands. yde@science.uva.nl

Frank Wolter
Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK. frank@csc.liv.ac.uk

Michael Zakharyaschev
School of Computer Science and Information Systems, Birkbeck College, Malet Street, London WC1E 7HX, UK. michael@dcs.bbk.ac.uk

PREFACE

MODAL LOGIC

This Handbook documents the current state of modal logic, a lively area of logical research which was born in philosophy, but which has since made its way into mathematics, linguistics, computer science, AI, and even economic game theory. As with other thriving scientific endeavours, it is not easy, and perhaps not even fruitful, to give an official definition of the subject. From the earliest days of modern modal logic, about a century ago now, there were many different interpretations, formalisms, and applications, and new developments have only added to this diversity. On the other hand, modal logic is also a remarkably coherent field in many ways, and its practitioners have no difficulty recognising research — and colleagues! — as being 'modal' in spirit. As editors, we see two broad perspectives that help give rise to this coherence. Writing with very broad strokes of the pen, we might say that the following two conceptions of the field have been particularly influential among modal logicians:

- *Modal logics as formalisations of modalities.* Many natural notions in language and science have a 'modal' character, in that they talk about possibility and necessity in some space of relevant situations. This was true for the original philosophical study of metaphysical modality, but it is equally true of modal logics of time, space, obligation, conditionality, knowledge, computation, and action, which have permeated other fields. Under this view, modal logics model natural reasoning concerning ubiquitous notions, and in doing so, they expand the descriptive scope of 'standard' logic. Technically, then, modal logics are obtained from standard logical systems (like classical propositional or predicate logic, intuitionistic logic, and so on) by adding new, *non-truth-functional operators* (that is, *modalities*). The non-truth-functional nature of the operators, reflecting the larger space of relevant situations, typically leads to systems richer than the underlying logic.

- *Modal logics as fragments of standard logics.* But surprisingly, another viewpoint on modal logic has become equally prominent among its practitioners. Under this view, modal logics inherit their semantics from the standard semantics of classical first or even higher-order predicate logic, but they restrict expressive power by using *operators instead of explicit quantification*. The term "fragment" carries no negative connotation of poverty here: curbing expressive power leads to systems with logical properties rather different from those of standard logic; the decidability of many modal logics is a striking example. The mathematical study of modal logics in this vein has brought to light a delicate balance between the expressive power and computational complexity of logical systems in general. That is, from this perspective modal logic is better viewed as a methodology for tapping into a core theme in standard logic. It is this second, more technical perspective, which provides much of the mathematical coherence of the field today.

The first perspective emphasises the *descriptive range* of modal logic as the study of key concepts and the reasoning patterns they give rise to. The second perspective emphasises the methodological aspect of *fine-structure*: modal languages bring to light the inner structure of classical systems. But these views are not in conflict. As the Handbook makes abundantly clear, most active research directions in modal logic take something from both. Indeed, the most widespread semantics of modal logic in terms of relational models (used in virtually every chapter of the Handbook) provides a setting in which these perspectives coexist fruitfully. Moreover, the two views help us better understand historical contributions made in the field. For example, the modal system **S4** was introduced in the 1920s as an analysis of the concept of necessary implication. But the limited expressive power of the formalism as an account of implicative structure turned out to be the key to **S4**'s wide range of other applications, and its attractive mathematical behaviour. In short, both perspectives on modal logic are widely applicable, and both have proved historically robust. Let's take a closer look at them, and see how they are related.

Modal logic as the study of old and new modalities Modal logic, conceived of as the *formal* study of modalities was invented in philosophy almost a century ago — though the *informal* study of modalities can be traced back much earlier: through the work of the medieval logicians, and back to the ancient Greeks. The first modal operators were introduced in order to solve the paradoxes of material implication and to obtain logics of necessity and possibility; the key figure here is C. I. Lewis, who published his pioneering work in 1918. Putting his idea in modern notation, we take some logical formula φ, and by prefixing it with a $\Box$ or a $\Diamond$ symbol we obtain the expressions $\Box\varphi$ ("the proposition φ is necessary") and $\Diamond\varphi$ ("the proposition φ is possible"). That is, the box and diamond notation enables us to assert fundamentally new modes of truth concerning the information expressed by φ, namely that it is necessary or possible.

In 1933, Kurt Gödel, driven by concerns in the foundations of mathematics, used modal operators to formalise the notion of mathematical provability. In particular, his work enabled intuitionistic logic to be reduced to classical logic extended with a provability operator, and the resulting logic turned out to be Lewis's system **S4**. A striking result indeed, but the general point is this: once again, modalities are being used to express fundamentally new modes of truth concerning a piece of information. In particular, now $\Box\varphi$ means that φ is provable, and $\Diamond\varphi$ means that it is consistent.

These early examples of applying modalities to logical formulas to make assertions concerning a novel mode of truth are only the tip of the iceberg. In the decades following the work of Lewis and Gödel, many modal operators were introduced and investigated, all dealing with truth in some space of possible situations. Tense logic (or temporal logic) arises with the addition of modalities like "eventually" or "earlier". Deontic logic adds modalities "it is permitted, or obligatory, that". In epistemic logic we make use of modalities like "it is known that", either for single agents or for groups. And conditional logics analyse further species of conditional reasoning far beyond Lewis's original account. This way of thinking about modal logic and modalities underlies the work of some of the field's most prominent pioneers, including G. H. von Wright, Arthur Prior, Jaakko Hintikka, Hans Kamp, and David Lewis.

Then the torch passed to other disciplines. In particular, temporal, dynamic and epistemic logics found their way into computer science, AI, and economic game theory. Temporal logics, of both branching and linear time, are now used in industry for automated verification of hardware and software. Epistemic, temporal and conditional operators are the main ingredient of knowledge-based programming. And modal logics of active agents with knowledge, beliefs, and desires form a theoretical backbone of modern accounts of intelligent dis-

tributed computing. Pioneers of modal methods in computer science include Edmund Clarke, Joe Halpern, Zohar Manna, Robin Milner, Rohit Parikh, Amir Pnueli, Vaughan Pratt, and many others — including quite a few of the authors and commentators in this Handbook. But again, diversity reigns, and creation of new modal formalisms for novel reasoning purposes continues unabated.

Summing up: it is entirely reasonable to say that modal logics are formalisms used to represent and reason about the plethora of modal notions that underlie, among other things, distributed computations and intelligent actions, and their corresponding modes of truth. They achieve this by making use of new operators, called modalities, whose truth-conditions involve access to some larger space of relevant situations, such as worlds, times, theories, or computational states. If you view modal logic in this descriptive way, you will be in excellent company.

Modal logic and the fine-structure of classical logics The invention of graph-based relational semantics (by Jaakko Hintikka, Stig Kanger, and Saul Kripke) in the late 1950s and early 1960s showed that standard modal logics could be regarded as fragments of first or second-order predicate logics. The underlying idea is straightforward. Suppose we read $\Box\varphi$ as "necessarily φ" and $\Diamond\varphi$ as "possibly φ". Drawing on an idea that dates back to the work of Leibniz, we could view "necessarily φ" as a claim that φ is true in all possible worlds, and "possibly φ" as a claim that φ is true in some possible world. Thus, modal operators perform quantification without making use of explicit variables and binding.

This idea, when expressed mathematically, has turned out to be the most significant milestone in the history of modal logic. For present purposes, the crucial idea in the above is just this. Because necessity means truth at *all* worlds, $\Box$ becomes linked to the universal quantifier $\forall$, and because possibility means truth at *some* world, $\Diamond$ becomes linked to the existential quantifier $\exists$. That is, necessity and possibility have been analysed in terms of classical quantification. The idea that modal operators are essentially concealed forms of classical quantification is fully general: for example, we can think of "eventually φ" as meaning "there is *some* future time at which φ holds", and we can think of "after performing a certain action, φ" as meaning "at *every* state which is accessible by performing a certain kind of action, φ holds".

But this is not all there is to the analogy. Viewed in this way, modal logics might just be different notation for classical ones! The creative difference is that the quantification in modal languages tends to be bounded in some way to 'relevant' or 'accessible' situations lying beyond the current one. In other words, we are working on structured universes of worlds, computational states, or what have you — and access is *mediated*. Together with the quantifier analogy, this bounded access explains two things. First, a number of properties of modal logics follow at once from those of their classical quantificational counterparts. Second, as the fragments of classical logic that modal operators correspond to typically have *less* expressive power than full first-order predicate logic, this results in many *new* properties. For example, the semantic invariances between models appropriate for modal expressive power are not those of classical logic, but rather turn out to be various forms of *bisimulation*, which preserve local properties of worlds and their transition patterns. Moreover, the study of fragments of classical systems and translations from modal to predicate logic (and sometimes back) has yielded satisfying explanations of why so many modal logics are decidable (unlike classical predicate logic), and why their computational complexity is often relatively low — and in a more activist way, modal analysis has led to the discovery of many new decidable fragments of classical logics. Indeed, the systematic design of modal languages stronger than the formalisms bequeathed to us by the founding fathers has been a major theme in recent research.

First-order languages have been the most traditional companions of modal ones, but the same points apply to modal fragments of higher-order languages, and — a significant development in recent years — to modal fragments of classical languages with fixed-point operators expressing iterative or recursive structures in action, computation, and knowledge representation.

Summing up: it is also entirely reasonable to say that modal logics are fragments of classical logics, which somehow strike an optimal balance between expressive power and computational simplicity. And if you view modal logic in this way (that is, as a laboratory for fine-structure) you will also be in excellent company.

Nowadays, few modal logicians would feel compelled to choose between the two perspectives just outlined. Their respective virtues are clear, and most researchers have assimilated both. Given the current explosion of practical applications and fundamental ideas in the field (amply documented in this Handbook), dogmatism concerning the nature of modal logic is becoming increasingly unsustainable. Let us emphasise this point a bit more.

Contemporary modal logic Modal logic today is a vast family of studies of modal notions, with the original philosophical and mathematical motivations still alive, but with an increasing symbiosis with other fields, and in particular, with computer science. Indeed, its interface with computer science (and more generally, informatics) is extremely broad, ranging from hardware and software verification, to ontologies in medical and bio-informatics, and the analysis of query languages for XML documents. Moreover, it also takes in commonsense reasoning in AI, covering issues ranging from representing and reasoning about space and time to modelling complex interactive multi-agent information systems. Nowadays, modal structures seem to occur everywhere, just as they did in the creative explosion of modality in the philosophical logic of the 1950s and 1960s. This independent (re-)discovery of modal operators in different settings is one of the strongest arguments for the stability and naturalness of the modal stance. Here are three striking illustrations of contemporary rediscoveries.

The first example is *description logic*, a branch of knowledge representation and reasoning in AI; nowadays it supplies many of the formalisms used to fix terminologies in medical and bio-informatics, and has been proposed as the language for annotating web pages to develop a semantic web. Since the late 1970s, the description logic community has articulated its fundamental research goal with great clarity: to obtain fragments of predicate logics which are computationally well-behaved but still have the expressive power required in knowledge representation applications. Intriguingly, around 1990 it became apparent that many of the logics obtained by pursuing this goal were in fact modal logics in a different notational guise. Bounded quantification turns out to be as fundamental for description logics as it is for standard modal logics. In this case, of course, the accessible objects are not worlds or situations but individual objects from some application domain (for example, the biological function of a DNA sequence). But many of the underlying ideas are the same, and this observation has opened the doors to joint work with the modal logic community, with benefits to both fields.

Another area where modal structure is currently surfacing is in the abstract study of *processes* in the emerging field of *coalgebra*. Though modally inspired process theories like dynamic logic, temporal logic, and process algebra have a long history in computer science, coalgebra adds a new twist. Starting from the work by Peter Aczel, Jon Barwise, and others on generalised set theory, coalgebra has now become a theory of finite and infinite processes, with deep connections with universal algebra and other parts of mathematics and theoretical computer science. The crucial feature here is that processes need not be bottom-up inductive, but can instead be top-down co-inductive streams of events. One gets to know a process through observation of events, chewing off the head of the event stream. The surprising discovery has been that such processes

and their observational analysis again shows clear modal patterns, leading to rapidly developing interfaces between coalgebra, modal logic, and universal algebra.

Finally, a third independent rediscovery of modal notions occurred in economic game theory. In the 1970s, Robert Aumann and others introduced formal models of *interactive knowledge* of agents in order to account for the reasoning underpinning the Nash equilibrium solutions that would be found by rational players of a game. Disregarding some differences in notation and style, the resulting formalisms turned out to be epistemic logics from the philosophical tradition, with operators for various forms of collective knowledge of groups. Over the past three decades, logical analysis of games has become another flourishing interface, with studies of beliefs and preferences in modal languages, and the development of dynamic logics of actions that can change modal attitudes as a game proceeds. Moreover, this modal study of games has now largely merged with that of computational processes, as games are naturally viewed as goal-driven multi-agent forms of computation. Traditional game theory was the mathematics of equilibrium, using methods from analysis and dynamical systems. The modal stance that is now emerging provides a natural level of fine-structure to go with this.

Despite this diversity of modal structures, there are also strong unifying tendencies, especially in the *mathematical metatheory* of the field, which got into its stride in the 1970s with work by Wim Blok, Kit Fine, Dov Gabbay, Rob Goldblatt, Larisa Maksimova, Steve Thomason, and others. Model theory of bisimulation and related frame constructions is one important strand here; among other things, it yielded broad definability techniques for matching modal languages with classical ones, and enabled the interpolation properties of modal languages to be charted. A second major strand is algebraic semantics and the duality between modal algebras and relational structures, which built on seminal work by Bjarni Jónsson and Alfred Tarski from the 1950s; since the rediscovery of their work in the 1970s, as universal algebra has grown in sophistication, so have its ties with modal logic. Another unifying force was the development of genuinely 'modal' techniques with wide applicability for proving completeness, axiomatizability and decidability results. And since the 1980s, further unifying mathematical themes have emerged: these include the study of the computational complexity of reasoning and its relation to succinctness; the exploration of the relation between various forms of tree automata, logical games, and the expressivity of modal languages; and the study of logic combinations and related model constructions, which has lead to a theory, still under active development, of various types of products of modal logics.

Another unifying tendency is the undeniable fact that the members of this growing modal family keep influencing each other. The word "applications" has a uni-directional ring to it, but it is a fact of life that every road can be walked both ways. For example, the action-oriented modal perspectives that are so prominent at the computer science interface have now crossed back into philosophy, giving rise to theories of information update and belief revision, which describe how agents come to acquire knowledge or change problematic beliefs. Moreover, setting up such systems also brings in conditional logics from the philosophical tradition; these are now seen as underlying belief revision and non-monotonic reasoning in deep and surprising ways.

This interdisciplinary and interactive setting is the stage where the drama of contemporary modal logic is played out. Modal structures are being studied in a growing number of areas, and often they seem to arise almost like naturally occurring phenomena; no premeditation by the modal logician is required. And at the same time, in response to this dramatic expansion, modal logicians have had to adopt a far wider range of technical ideas and tools than ever before, tools that lie beyond the placid waters of traditional textbook introductions to the field. It was against this exciting and challenging background that this Handbook was conceived.

THE HANDBOOK OF MODAL LOGIC

This Handbook presents a detailed overview of the main lines of research in contemporary modal logic. The editors have tried to present a fair picture of the modern scene, and one that (to the extent possible in a one-volume handbook) reflects the scene in its entirety. Moreover, the selection of authors has been made with a view toward representing the most active and creative research communities worldwide

The tricky question is: what is the best compromise between "detailed" and "overview"? We felt the field would be best served by a single volume handbook; that is, we opted for judicious selection and bounded access. Not that it would have been difficult to design a multi-volume handbook. On the contrary, the most frequent request we had from our authors was for more generous pages limits; the pull towards detail is strongly felt in a field such as modal logic, and quite rightly so. Many of the most treasured results and insights of the field are the results of years of painstaking work. In a sense, every student of the subject has to retrace these intellectual journeys; short cuts aren't possible.

But the evident need for an accessible overview of the whole, leaving deeper access to valleys and caves to a second stage, suggests that the one volume choice was correct. One of the points that emerged most strongly during the Handbook's preparation was just how unified modal logic still is. To be sure, some of its branches are now highly technical, whereas other branches are better thought of as conceptual investigations which use the language of modal logic as an aid to precision. Furthermore, some work emphasises generality and takes mathematical criteria as its primary guide, whereas other areas may be highly specific in their focus and take their cue from applications. But in spite of such differences, the field remains stubbornly coherent, and surprisingly comprehensible. It is not an exaggeration to claim that most researchers in modal logic have at least a nodding acquaintance with the majority of the topics discussed in this Handbook, feel that this sort of extensive knowledge is useful, and would like their students to have a map of the terrain at least as wide. This Handbook is an attempt to provide such a map. To put it another way, it tries to gather together the background assumptions, the working knowledge, the mathematical techniques, and the general world view that add up to that somewhat elusive entity "contemporary modal logic", and to bring it together in a digestible form. We hope it will provide exactly the sort of snapshot of the field that will serve as intellectual nourishment for the next generation of researchers in, and users of, modal logic.

We made no serious effort to impose notational or other kinds of uniformity on the authors. This was partly for pragmatic reasons: it was always evident that the attempt to impose a standard notation would please nobody — and who is to say that linguistic diversity is worse than linguistic, or even cultural, uniformity? But there are deeper reasons for our hands-off stance. Research in contemporary modal logic takes place in a shifting environment that jostles the borders of many fields. One modal logician's interests may lead to the frontiers of linguistics or automated reasoning, another to the foundations of computation and games, and another towards purely mathematical issues concerning topological spaces. Moreover, these interests keep converging, and diverging, in new and often unpredictable ways; we have given some telling examples already. Had this Handbook appeared five years earlier, the borders between various fields would have been drawn somewhat differently, and it is entirely possible that in another five years many will have to be drawn yet again. In the face of such flux, imposing uniformity would be an artificial exercise. Modal logic is a sea, and the point of this Handbook is to help the reader learn to swim in it; pretending it is a swimming pool distorts the truth, and what's worse, is unhelpful.

USING THE HANDBOOK

We suspect that most of our readers will be accustomed to navigating their way through weighty research tomes, and will require little in the way of advice. In particular, readers who already know something about modal logic should simply consult the Table of Contents and start where it looks most interesting; while there are cross-references between the chapters, each is, to a great extent, self contained, so this is a viable strategy. Moreover, the Handbook can be used as a reference to the field. In particular, the index gives detailed entries for most common logics, notions, and results.

But we have also attempted to make the Handbook accessible to less experienced readers. Now, we should say right away that by "less experienced" we mean less experienced in *modal* logic. This is a technical volume, and readers without technical background are going to find it difficult. Thus our less experienced reader is someone who already has some understanding of what modern logic is about, and why and where it is useful, and who wants to find out something about what modern modal logic has to offer.

The Handbook is structured to provide an answer to such readers. The book has 21 chapters, and is divided into four parts: Basic Theory, Advanced Theory, Variations and Extensions, and Applications. Although independent, together they tell a story, the story of contemporary modal logic. This story starts with the basic tools and techniques, takes the reader to the outer reaches of the underlying mathematical theory, surveys the key points where the modal approach is being adapted and extended, and finishes by examining the various applications which modal logic serves and from which it draws inspiration. Let's take a closer look at how all this unfolds over the course of the Handbook.

Part 1. Basic Theory The chapters in Part 1 lay the foundation for later ones. Together they present an overview of the most fundamental themes, techniques and results in contemporary modal logic. The growing impact of computer science is clearly reflected in the choice of topics: two chapters are wholly devoted to complexity, decision methods, and implementation.

Chapter 1. Modal Logic: A Semantic Perspective. Patrick Blackburn and Johan van Benthem.

This chapter discusses the semantic ideas underlying modern modal logic, and in particular, Kripke semantics — or relational semantics, as it now (more informatively and fairly) tends to be called. It introduces the basic model theoretic constructions in a modern way, explores links between modal logic and classical (predicate) logic, both on models and on frames, and examines the extent to which the key semantic ideas transfer to richer modal logics and languages while maintaining a relatively low computational complexity. It also introduces some alternative viewpoints: algebraic semantics, neighbourhood semantics, and topological semantics.

Chapter 2. Modal Proof Theory. Melvin Fitting.

Modal proof theory is the study of syntactic calculi, defined in terms of symbol manipulation, for performing modal logical reasoning. How can such systems be designed? Is there an interesting range of design choices? And how can the syntactic ideas underlying proof calculi be linked with the semantic ideas introduced in Chapter 1? This chapter answers these questions by introducing a wide range of proof styles, and discussing modal completeness theory, the fundamental bridge between proof-theoretical and semantic investigations.

Chapter 3. Complexity of Modal Logic. Maarten Marx.

The basic modal language, when interpreted over relational models, can be regarded as a decidable fragment of classical logic. But this observation immediately leads to a host of further questions. Given that it is decidable, how difficult is it to compute with? That is, what is the computational complexity of determining validity, or of performing more modest tasks like model checking? And what are the parameters that affect modal complexity results, and what happens when we play with their settings? This chapter, an introduction to the computational complexity of modal logic, provides some fundamental answers.

Chapter 4. Computational Modal Logic. Ian Horrocks, Ullrich Hustadt, Ulrike Sattler, and Renate Schmidt.

Although Chapter 2 introduced modal proof theory, and Chapter 3 studied the computational complexity of modal logic, only with this chapter do we reach the heartland of computational modal logic: how to build modal inference systems that are efficient in practice. Although it surveys a number of topics, this chapter concentrates on two fundamental issues: how resolution and tableaux methods can be adapted to modal logic, and how these methods are related.

Part 2. Advanced theory The chapters in Part 2 provide a deep and wide ranging theoretical analysis of modal logic that is broad enough to apply to many application areas. In some cases they provide deeper perspectives on topics already introduced in Part 1, but often they introduce ideas barely hinted at in earlier chapters. Taken together, they present the central core of contemporary insight into the mathematical structure of modal logic.

Chapter 5. Model Theory of Modal Logic. Valentin Goranko and Martin Otto.

At the heart of relational semantics is the idea of interpreting modal languages over relational structures by viewing them as fragments of first-order predicate logic or some stronger formalism. This perspective is not only intuitively attractive, it also makes available to modal logic the results and tools developed in such areas as classical model theory and finite model theory. This chapter shows, in great detail, how such tools can be put to work to gain a deep mathematical understanding of modal model theory, and what makes it *sui generis*.

Chapter 6. Algebras and Coalgebras. Yde Venema.

This chapter develops in detail the algebraic semantics of modal logic and introduces an alternative coalgebraic approach. Algebraic semantics, which has thrived as a research area since the early 1970s, is important because it makes it possible to apply general techniques from universal algebra to the study of modal logic. The approach has given rise to some of the most penetrating analyses of the mathematics of modality. The more recent coalgebraic approach, which also links up with category theory, is valuable because it offers a uniform mathematical setting in which to analyse dynamic systems in terms of modal logic.

Chapter 7. Modal Decision Problems. Frank Wolter and Michael Zakharyaschev.

Modal logic is decidable — or at least it is when interpreted on the class of all models. But change the interpreting class of models, and you change the logical validities, and decidability is typically lost when the structural conditions come too close to ‘danger zones’ such as tiling patterns, arithmetic, or other structures allowing for Turing machine computation. This chapter is a detailed examination of how such properties as decidability, the finite model property, and finite axiomatisability are distributed across the lattice of normal modal logics. The emphasis is on providing general results, and drawing attention to important open questions.

Chapter 8. Modal Consequence Relations. Marcus Kracht.

The notion of consequence, which tells us when a conclusion follows from given premises, is a fundamental logical concept, and in the setting of modal logic it can be defined in a number of different ways. This chapter surveys some of the most important ideas, covering in detail such topics as local versus global consequence, reducing multimodal consequence to monomodal consequence, interpolation theorems, and the admissibility of rules. As with Chapter 7, the emphasis is on providing general results which apply across a wide range of logics.

Part 3. Variations and Extensions The main focus of the chapters in Parts 1 and 2 was on relatively simple propositional modal systems based on (collections of) $\Diamond$ and $\Box$ modalities. Such systems are historically central but they don't exhaust the kinds of logic that now go under the name "modal logic". The chapters in Part 3 introduce some of the extensions and variations of the basic modal technology that the reader is likely to encounter.

Chapter 9. First-order Modal Logic. Torben Braüner and Silvio Ghilardi.

First-order modal logics are modal logics in which the underlying propositional logic has been replaced by a first-order predicate logic. These are one of the oldest forms of modal logic, and arguably the most philosophically important. They also pose some of the most difficult mathematical challenges. This chapter first surveys basic first-order modal logics, and then examines recent attempts to find a general mathematical setting in which to analyse them.

Chapter 10. Higher-order Modal Logic. Reinhard Muskens.

The basic ideas of modal logic have also been extended to higher-order settings, and indeed, extended in a number of different ways. This chapter motivates such extensions, some of them from linguistic semantics in the tradition of Richard Montague, examines some of the more historically influential ones, indicates some of the difficulties that can arise in the transition to higher-order logic, and finally shows how these difficulties can be overcome.

Chapter 11. Temporal Logic. Ian Hodkinson and Mark Reynolds.

Temporal logic is one of the classic branches of modal logic and is currently one of the most active. It has been remarkably fruitful in the issues it has raised (what kinds of temporal structure should we work with?), the results it has given rise to (it is the source of some of the most interesting expressivity results in modal logic), and as an applied tool (contemporary model checking technology is based on temporal logic). This chapter will introduce the reader to the key issues of this important and diverse area.

Chapter 12. Modal μ-Calculi. Julian Bradfield and Colin Stirling.

In the late 1960s, pioneers in reasoning about programs adopted some key ideas of modal logic. They repaid the debt handsomely. Among other things, they developed dynamic logic (used in several chapters of this Handbook), and the modal μ-calculus, one of the most interesting modal formalisms to have emerged in the last two decades. This provides second-order expressive power sufficient to generalise the most common temporal logics, but is still decidable and has the finite model property. It raises many intriguing issues about the interface between modal logic, complexity theory, and automata theory.

Chapter 13. Description Logic. Franz Baader and Carsten Lutz.

Modal logic is sometimes thought of as an intrinsically intensional logic, suitable only for applications such as reasoning about necessity, possibility, and knowledge. But description logics (which developed from pioneering work in the AI community) are undeniably modal logics and, as the description logic community has shown in impressive detail, are extremely well suited for

reasoning about ordinary individuals and the relations between them. This chapter is a detailed introduction to one of modal logic's closest neighbours.

Chapter 14. Hybrid Logics. Carlos Areces and Balder ten Cate.

Standard modal logics use modalities for talking about the relations in relational structures, but don't contain mechanisms for talking about particular worlds. Hybrid logic arises when mechanisms for naming and asserting identity of worlds are added; to give an analogy, they are to standard modal systems what first-order languages with equality are to equality-free languages. This chapter surveys the proof theory, expressivity, and complexity of a number of the better known hybrid logics, thereby giving a snapshot of the logical territory lying between the basic modal languages and their classical companions.

Chapter 15. Combining Modal Logics. Agi Kurucz.

The idea of combining modal logics (for example, a modal logic of time with a modal logic of knowledge) is natural for many applications. But how can modal logics be combined, and what happens when you combine them? This chapter surveys two key combination methods (fusions and products) in detail, shows how various properties do (or do not) transfer from the individual logics to the combination, and briefly examines a number of other combination methods. The properties of combined logics turn out to depend in subtle ways on those of their components plus the particular method of combination.

Part 4. Applications Historically, modal logic has been profoundly influenced by its applications, which have been extremely diverse in nature. The chapters in Part 4 survey the key application domains, thereby showing where modal logic comes from, where it has visited along the way, and also indicating areas to which it is likely to return.

Chapter 16. Modal Logic in Mathematics. Sergei Artemov.

Mathematics is one of modal logic's oldest application areas. In particular, the pioneering work of Gödel in the 1930s showed that modal logic offered an important perspective on the notion of mathematical provability, and (more recently) modal logics of proof have been developed. But modal logic also gives rise to natural logics of space and dynamic systems, and even turns out to be a tool with applications in set theory. This chapter surveys these themes. In doing so, it emphasises an intriguing duality in interpreting the modal box: either as the universal quantifier "in all worlds", or as the existential "there exists a proof". Just when and how such accounts converge is a deep metamathematical issue.

Chapter 17. Automata-Theoretic Techniques for Temporal Reasoning. Moshe Y. Vardi.

Many modal and temporal logics can be viewed as fragments of monadic second-order logic over trees in a suitable signature, so there is a clear theoretical link (via Rabin's celebrated decidability theorem) between modal logic and automata theory. But this link turns out to have practical repercussions for computational applications. In particular, by viewing temporal formulas as giving rise to what are known as "alternating automata", we gain a theoretically transparent but also practical perspective on both validity and model checking, one of the most significant applications of contemporary modal logic.

Chapter 18. Intelligent Agents and Common-Sense Reasoning. John-Jules Meyer and Frank Veltman.

Modal logics have been used in AI in a number of different ways. This chapter discusses two of its more important roles there. The first is as a logic of agents, and here the chapter takes the reader from basic epistemic and deontic logic to multi-agent logics of beliefs, desires, and

intentions. The second is as a model of common sense reasoning, and here the chapter covers modal treatments of counterfactual conditionals and non-monotonic reasoning in a variety of guises, including default reasoning.

Chapter 19. Applications of Modal Logic in Linguistics. Lawrence S. Moss and Hans Jörg Tiede.

Modal logic is best known in linguistics for the light it throws on semantics; indeed Richard Montague's use of higher-order modal logic for this purposes is widely considered to be the starting point of modern natural language semantics. Recently, however, modal logic has also been used to analyse syntactic structure, and interesting links with formal language theory have thereby emerged. This chapter discusses both topics, providing a sophisticated view of modern interfaces between logic and natural language.

Chapter 20. Modal Logic for Games and Information. Wiebe van der Hoek and Marc Pauly.

Game-theoretic ideas have long played an influential rule in analysing various branches of logic, but the focus of this chapter is on using modal logics to describe and reason about games. After introducing the basic ideas of game theory, it systematically investigates how modal logic can be used to do this. Three main topics are discussed: modeling imperfect information and multi-agent information update via dynamic epistemic logics; reasoning about game structure through operations for combining games; and logics of collective action and the power of coalitions of agents over time.

Chapter 21. Modal Logic and Philosophy. Sten Lindström and Krister Segerberg.

Modal logic was born in philosophy, and though it has since travelled widely, it still retains important links with the discipline. This chapter first discusses the historical heartland of philosophical modal logic: namely, the scope and limitations of modal logic as an account of necessity and possibility. It then examines two more recent topics: modal logic and the logic of belief change, and modal logic as a logic of action.

Together, these chapters present a broad picture of modal logic today. Of course, some choices had to be made, and some bias may remain. In particular, the emphasis throughout has been on relational graph-style models. The editors fully acknowledge that there are other important traditions, such as modal logics based on non-classical logics, proof-theoretic semantics, as well as the more general neighbourhood semantics. Some of these have a historical pedigree reaching back to the 1930s, and they are still very much alive. These approaches do occur at various places in the book, but we have not made them fundamental to the Handbook's architecture. We think this is a fair reflection of the bulk of current research, but times may change.

Some readers may also think that this Handbook has a bias toward propositional model systems, leaving predicate-logical versions underrepresented. This may be true to some extent — however, the chapters on modal predicate logic, modal logic in philosophy, and also temporal logic give a clear account not only of the basic theory of modal predicate logic, but also of recent developments of its mathematics and its relevance to computer science. But there is also a more defiant stance. Predicate logic itself can be profitably viewed as a modal logic of variable assignment and assignment change. And in that light, modal predicate logic is not some privileged enrichment of modal logic. It is really about combining a propositional modal logic of worlds and one of variable assignment. And the reader can learn a lot about both, and about combination methods, from the pages of this Handbook!

Once these chapters are seen together, many questions may at once be formulated concerning further relationships between them. The editors considered adding some remarks on this topic,

as diversity tempered by the resulting need for system comparison are a major driving force for innovation in the field. But in the end, we have decided to let the chapters, and their juxtaposition, speak for themselves.

Finally, we would like to remark that modal logic is as much a community of living people as a family of systems. The results and insights in this Handbook exist only because of a long line of distinguished researchers who have shaped modal logic and its interfaces with computer science and other fields. The Hall of Fame of our field certainly extends beyond the grand old names of the classical period; just read this Handbook, and you will come to know the grand new names by their fruits.

COMMENTATORS

Following an idea of Dov Gabbay's used in many publications, we made use of the designated commentator system in this Handbook. That is, in addition to editorial feedback, we attempted to find, for each chapter, a reader who could provide the kind of feedback that would inspire the authors during the writing process. In some cases we chose commentators with special expertise in the topic of the chapter. In other cases, we felt that the comments offered by someone working in a somewhat different area might be more appropriate and helpful. Moreover, whenever possible, we chose authors of other chapters as commentators, as we felt this would improve the Handbook's coherence.

We are extremely grateful to everyone who agreed to undertake this task; in many cases the input of a commentator acted as precisely the catalyst needed to help the full potential of a chapter to emerge. Our commentators were (Chapter 19 had no commentator):

Chapter 1: Aleksander Chagrov.	*Chapter 2: Heinrich Wansing.*
Chapter 3: Martin Otto.	*Chapter 4: Carlos Areces.*
Chapter 5: Maarten de Rijke.	*Chapter 6: Aleksander Kurz.*
Chapter 7: Rosalie Iemhoff.	*Chapter 8: Ian Hodkinson.*
Chapter 9: Melvin Fitting.	*Chapter 10: Grigori Mints.*
Chapter 11: Valentin Goranko.	*Chapter 12: Yde Venema.*
Chapter 13: Silvio Ghilardi.	*Chapter 14: Ulrike Sattler.*
Chapter 15: Carsten Lutz.	*Chapter 16: Rineke Verbrugge.*
Chapter 17: Julian Bradfield.	*Chapter 18: Reinhard Muskens.*
Chapter 20: Giacomo Bonanno.	*Chapter 21: Vincent Hendricks.*

FURTHER INFORMATION

We have set up a home page for the Handbook at:

`http://www.csc.liv.ac.uk/~frank/MLHandbook`

We will make available there any corrections that may need to be made, and news concerning future Handbook-related developments. We welcome feedback from our readers.

It would not be useful to attempt to list all the workshops, conferences, and journals where work on modal logic is published; given what we have said about its wide range of applications and techniques, it will come as no surprise that such work may be made public in a wide variety of forums. However it is worthwhile taking this opportunity to mention two workshops specifically

devoted to modal logic. The first is *Advances in Modal Logic* (AiML), modal logic's main event, which is held every two years. You can find out more about this event, and the associated book series at:

`http://www.aiml.net`

Here we'll simply say that AiML attempts to bring together scholars working in all areas of modal logic and its applications. The second workshop is *Methods for Modalities* (M4M), see

`http://m4m.loria.fr`

M4M is also held every two years. It is more practically oriented than AiML, focusing on the development of computational tools and results for modal logic.

ACKNOWLEDGEMENTS

Editing a handbook is a lengthy task, and sometimes, very modally, it seems as if all the things that could go wrong take the opportunity to actually do so. Fortunately, we had the benefit of a great deal of support, which generally enabled us to dissolve problems before they became too pressing. Indeed, in retrospect, it's clear we had a relatively easy time of it, and we would like to thank all the people who made this possible.

First, and foremost, we would like to thank the authors of the Handbook's chapters. They showed enormous enthusiasm for the project, many of them going out of their way to attend a meeting to set the Handbook in motion (in March 2004, Amsterdam; the meeting was kindly sponsored by NWO, KNAW and other Dutch scientific organisations). We received a great deal of valuable feedback from them, and they showed good-humoured patience when faced by style file changes, demands for quick attention to page proofs, and requests for additions or alterations to their chapters. It is a privilege to work with such a lively and cooperative community. The resulting Handbook is a testimony to their efforts. Our deepest thanks to them all.

We were also extremely lucky with the support we received on the production side. First, our thanks to Arjen Sevenster, of Elsevier, for his enthusiastic support in making this Handbook happen right from the start, including judicious alternations of leaving us be and prodding, and to Andy Deelen, our contact at Elsevier, for her prompt responses to our questions. We are also extremely grateful to Carlos Areces, Christian Günsel, Eric Kow, Oliver Kutz and Dirk Walther for the technical help they gave us with the website, the style files, and the indexing. Special thanks are due to Balder ten Cate for organising the meeting in Amsterdam, and to Guido Governatori, who wrote the LaTeX style file used to produce the final version of the Handbook, and who responded quickly and generously to our urgent requests.

And, finally, we come to Jane Spurr. Without her, it is hard to imagine how we could have put the Handbook together. It was Jane who grafted the 21 chapters into one big LaTeX file overcoming many evil obstacles, who dealt with final corrections, and who found ways of making our numerous formatting requests work. Thanks to her unflagging good humour and patience, the final production stage became a positive pleasure.

Patrick Blackburn
Johan van Benthem
Frank Wolter

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

1

MODAL LOGIC: A SEMANTIC PERSPECTIVE

Patrick Blackburn and Johan van Benthem

1	Introduction	2
2	Basic modal logic	3
2.1	First steps in relational semantics	3
2.2	The standard translation	10
3	Bisimulation and definability	12
3.1	Drawing distinctions	12
3.2	Bisimulation	13
3.3	Invariance and definability in first-order logic	18
3.4	Invariance and definability in modal logic	18
3.5	Modal logic and first-order logic compared	21
3.6	Bisimulation as a game	23
4	Computation and complexity	24
4.1	Model checking	24
4.2	Satisfiability and validity: decidability	27
4.3	Satisfiability and validity: complexity	30
4.4	Other reasoning tasks	31
5	Richer logics	33
5.1	Axioms and relational frame properties	34
5.2	Frame definability and undefinability	35
5.3	Frame correspondence and second-order logic	38
5.4	First-order frame definability	40
5.5	Correspondence in richer languages	41
5.6	Remarks on computability	43
6	Richer languages	45
6.1	The universal modality	46
6.2	Hybrid logic	49
6.3	Temporal logic with Until and Since operators	52
6.4	Conditional logic	54
6.5	The guarded fragment	56
6.6	Propositional Dynamic Logic	58
6.7	The modal μ-calculus	61
6.8	Combined logics	64
6.9	First-order modal logic	66
6.10	General perspectives	68
7	Alternative semantics	72
7.1	Algebraic semantics	72
7.2	Neighbourhood semantics	75
7.3	Topological semantics	76
8	Modal logic and its changing environment	79

1 INTRODUCTION

This chapter introduces modal logic from a semantic perspective. That is, it presents modal logic as a tool for talking about *structures* or *models*. But what kind of structures can modal logic talk about?

There is no single answer. For example, modal logic can be given an *algebraic semantics*, and under this interpretation modal logic is a tool for talking about what are known as boolean algebras with operators. And modal logic can be given a *topological semantics*, so it can also be viewed as a tool for talking about topologies. But although we briefly discuss algebraic and topological semantics, for the most part this chapter focuses on modal logic as a tool for talking about *graphs*. To put it another way, this chapter is devoted to what is known as the *relational* or *Kripke* semantics for modal logic. This is the best known and (with the possible exception of algebraic semantics) the best explored style of modal semantics. It is also, arguably, the most intuitive. Over the years modal logic has been applied in many different ways. It has been used as a tool for reasoning about time, beliefs, computational systems, necessity and possibility, and much else besides. These applications, though diverse, have something important in common: the key ideas they employ (flows of time, relations between epistemic alternatives, transitions between computational states, networks of possible worlds) can all be represented as simple graph-like structures. And as we shall see, modal logic is an interesting tool for talking about such structures: it provides an internal perspective on the information they contain.

But modal logic is not the only tool for talking about graphs, and this brings us to one of the major themes of the chapter: the relationship between modal logic and other forms of logic. As we shall see, under the graph-based perspective discussed here, modal logic is closely linked to both first- and second-order classical logic. This immediately raises interesting questions. How does modal logic compare with these logics as a tool for talking about graphs? Can modal expressivity over graphs be characterised in terms of classical logic? We shall ask (and answer) such questions in the course of the chapter.

Games (in various guises) are another recurring motif. The simple way that modal formulas are interpreted on graphs naturally gives rise to games and game-like concepts. The most important of these is the notion of *bisimulation*. This is a relation between two models, weaker than isomorphism, which can be thought of as giving rise to a transition-matching game between two players. As we shall see, this concept holds the key to modal model theory and characterises the link with first-order logic.

This chapter has two pedagogical goals. The first is to provide a bread-and-butter introduction to relational semantics for modal logic that can be used as a basis for tackling the more advanced chapters in this handbook. Thus the reader will find here definitions and discussions of all the basic tools needed in modal model theory (such as the standard translation, generated submodels, bounded morphisms, and so on). Basic results about these concepts are stated and some simple proofs are given. But we have a second, more ambitious, goal: to help the reader start thinking semantically. We want to give the reader a sense of how modal logicians view structure, and what they look for when exploring new logics. To this end we have tried to isolate the intuitions that guide working modal logicians, and to present them vividly. We also make numerous asides, some of which touch on advanced logical topics. Their purpose is to situate the key ideas in a wider context, and even beginners should try to follow them.

Here is our plan. In Section 2, we introduce basic modal languages and the graphs over which they are interpreted. We give the satisfaction definition (which tells us how to interpret modal formulas in graphs) and the standard translation (which links modal logic with classical logic).

With these preliminaries out of the way, we are ready to go deeper. What can (and cannot) modal languages say about graphs? In Section 3 we introduce the notion of bisimulation and use it to develop some answers; among other things, we characterise modal logic as a fragment of first-order logic. In Section 4 we examine the computability and computational complexity of modal logic. A shift of topic? Not at all. In essence, this section examines modal logic as a tool for talking about *finite* graphs. In Section 5 we move to the level of frames and re-examine the link between modal and classical logic. As we shall see, at this level the fundamental correspondence is between modal logic and (monadic) second-order logic. In Section 6 we move beyond the basic modal language and discuss a number of richer languages that offer more expressivity. But what makes them all modal? As we shall see, many of the themes explored in earlier sections re-emerge, and point towards an idea that seems to lie at the heart of modal logic: guarding. Moreover, in some cases it is possible to prove Lindström-style characterisation results. In Section 7 we discuss three alternatives to relational semantics, namely algebraic, neighbourhood, and topological semantics. We conclude in Section 8.

Two final remarks. First, although we introduce modal logic from scratch, we assume that the reader has at least a basic understanding of classical first-order logic (especially its model-theoretic semantics) and some grasp of the notion of computability. Any standard introduction to mathematical logic (Enderton [37] is a good choice) supplies more than enough material to follow the main line of the chapter. Second, we *don't* discuss modal proof-theory or related notions such as completeness in any detail (these topics are the focus of Chapter 2 of this handbook). Although we haven't banished all mention of normal modal logics and completeness from the chapter, in our view traditional introductions to modal logic tend to overemphasise these topics. We want this chapter to act as a counterbalance. As we hope to convince the reader, simply asking the question "But what can I *say* with these languages?" swiftly leads to interesting territory.

2 BASIC MODAL LOGIC

In this section we introduce the basic modal language and its relational semantics. We define basic modal syntax, introduce models and frames, and give the satisfaction definition. We then draw the reader's attention to the internal perspective that modal languages offer on relational structure, and explain why models and frames should be thought of as graphs. Following this we give the standard translation. This enables us to convert any basic modal formula into a first-order formula with one free variable. The standard translation is a bridge between the modal and classical worlds, a bridge that underlies much of the work of this chapter.

2.1 First steps in relational semantics

Suppose we have a set of proposition symbols (whose elements we typically write as p, q, r and so on) and a set of modality symbols (whose elements we typically write as m, m', m'', and so on). The choice of PROP and MOD is called the *signature* (or *similarity type*) of the language; in what follows we'll tacitly assume that PROP is denumerably infinite, and we'll often work with signatures in which MOD contains only a single element. Given a signature, we define the *basic modal language* (over the signature) as follows:

$$\varphi \quad ::= \quad p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \varphi \leftrightarrow \psi \mid \langle m \rangle \varphi \mid [m]\varphi.$$

That is, a basic modal formula is either a proposition symbol, a boolean constant, a boolean combination of basic modal formulas, or (most interesting of all) a formula prefixed by a diamond

or a box. There is redundancy in the way we have defined basic modal languages: we don't need all these boolean connectives as primitives, and it will follow from the satisfaction definition given below that, for all $m \in \text{MOD}$, $[m]\varphi$ is equivalent to $\neg\langle m\rangle\neg\varphi$ and $\langle m\rangle\varphi$ is equivalent to $\neg[m]\neg\varphi$ (so boxes and diamonds are what are known as *dual connectives*, just as $\exists$ and $\forall$ are in first-order logic). But we won't bother picking out a preferred set of primitives, as this is not relevant to our discussion. If there is only one modality in our language (that is, if MOD has only one element) we simply write $\Diamond$ and $\Box$ for its diamond and box forms. We often tacitly assume that some signature has been fixed, and say things like "the basic modal language", or "the basic modal language with one diamond". We won't need many syntactic concepts in this chapter, but the following ones will be useful. First, the *subformulas* of a basic modal formula φ are φ itself together with all the formulas used to build φ. Second, we say that a subformula ψ of φ occurs *positively* if it is under the scope of an even number of negations, otherwise we say it occurs *negatively* (when this definition is applied, subformulas of the form $\psi \to \theta$ should be read as $\neg\psi \vee \theta$, and subformulas of the form $\bot$ should be read as $\neg\top$). Finally, the *modal operator depth* of a basic modal formula φ is the maximum level of nesting of modalities in φ, and we write *md*(φ) to denote this number.

A *model* (or *Kripke model*) $\mathfrak{M}$ for the basic modal language (over some fixed signature) is a triple $\mathfrak{M} = (W, \{R^m\}_{m\in\text{MOD}}, V)$. Here W, the *domain*, is a non-empty set, whose elements we usually call *points*, but which, for reasons which will soon be clear, are sometimes called *states*, *times*, *situations*, *worlds* and other things besides. Each R^m in a model is a binary relation on W, and V is a function (the valuation) that assigns to each proposition symbol p in PROP a subset $V(p)$ of W; think of $V(p)$ as the set of points in $\mathfrak{M}$ where p is true. The first two components $(W, \{R^m\}_{m\in\text{MOD}})$ of $\mathfrak{M}$ are called the *frame* underlying the model. If there is only one relation in the model, we typically write (W, R) for its frame, and (W, R, V) for the model itself. We encourage the reader to think of Kripke models as graphs (or to be slightly more precise, *directed graphs*, that is, graphs whose points are linked by directed arrows) and will shortly give some examples which show why this is helpful.

Suppose w is a point in a model $\mathfrak{M} = (W, \{R^m\}_{m\in\text{MOD}}, V)$. Then we inductively define the notion of a formula φ being *satisfied* (or *true*) in $\mathfrak{M}$ at point w as follows (we omit some of the clauses for the booleans):

$$
\begin{array}{rll}
\mathfrak{M}, w \models p & \text{iff} & w \in V(p), \\
\mathfrak{M}, w \models \top & & \text{always}, \\
\mathfrak{M}, w \models \bot & & \text{never}, \\
\mathfrak{M}, w \models \neg\varphi & \text{iff} & \text{not } \mathfrak{M}, w \models \varphi \text{ (notation: } \mathfrak{M}, w \not\models \varphi), \\
\mathfrak{M}, w \models \varphi \wedge \psi & \text{iff} & \mathfrak{M}, w \models \varphi \text{ and } \mathfrak{M}, w \models \psi, \\
\mathfrak{M}, w \models \varphi \to \psi & \text{iff} & \mathfrak{M}, w \not\models \varphi \text{ or } \mathfrak{M}, w \models \psi, \\
\mathfrak{M}, w \models \langle m\rangle\varphi & \text{iff} & \text{for some } v \in W \text{ such that } R^m wv \text{ we have } \mathfrak{M}, v \models \varphi, \\
\mathfrak{M}, w \models [m]\varphi & \text{iff} & \text{for all } v \in W \text{ such that } R^m wv \text{ we have } \mathfrak{M}, v \models \varphi.
\end{array}
$$

A formula φ is *globally satisfied* (*globally true*) in a model $\mathfrak{M}$ if it is satisfied at all points in $\mathfrak{M}$, and if this is the case we write $\mathfrak{M} \models \varphi$. A formula φ is *valid* if it is globally satisfied in all models, and if this is the case we write $\models \varphi$. A formula φ is *satisfiable in a model* $\mathfrak{M}$ if there is some point in $\mathfrak{M}$ at which φ is satisfied, and φ is *satisfiable* if there is some point in some model at which it is satisfied. These definitions are lifted to sets of formulas in the obvious way. For

example, a set of basic modal formulas Σ is satisfiable if there is some point in some model at which all the formulas it contains are satisfied. A formula φ is a *semantic consequence* of a set of formulas Σ if for all models $\mathfrak{M}$ and all points w in $\mathfrak{M}$, if $\mathfrak{M}, w \models \Sigma$ then $\mathfrak{M}, w \models \varphi$, and in such a case we write $\Sigma \models \varphi$. Instead of writing $\{\varphi\} \models \psi$ we write $\varphi \models \psi$.

We now have all the concepts needed to begin exploring modal logic. But instead of moving on, let us reflect upon the ideas just introduced. First, note the *internal* character of the modal satisfaction definition: *modal formulas talk about Kripke models from the inside*. In first-order classical logic, when we talk about a model, we do so from the outside. A *sentence* of first-order logic does not depend on the contextual information contained in assignments of values to variables: sentences take a bird's-eye-view of structure, and, irrespective of the variable assignment we use, are simply true or false of a given model. Modal logic works differently: we evaluate formulas *inside* models *at some particular point*. A modal formula is like an automaton placed inside a structure at some point w, and forced to explore by making transitions to accessible points. This may seem a fanciful way of thinking about the satisfaction definition, but it turns out to be crucial. When we isolate the mathematical content of this intuition, we are led, fairly directly, to the notion of *bisimulation*, the key to modal model theory, which we will introduce in Section 3.

Second, note that basic modal languages are syntactically extremely simple: we are working with languages of propositional logic augmented with additional unary operators. And yet these languages clearly pack quantificational punch. Diamonds and boxes can be thought of as macros that encode quantification over R^m-accessible states in a perspicuous variable-free notation. We will shortly define the *standard translation*, which makes this macro analogy precise.

Third, note that Kripke models can (and in our opinion should) be thought of as (directed) graphs. As we have already mentioned, modal logic has been applied in many different areas. What these areas have in common is that they deal with applications in which the important ideas can be represented by relatively simple graph-like structures. Let's consider some examples,

A classic interpretation of Kripke models of the form (W, R, V) is to regard the elements of W as times, and the relation R as the relation of temporal precedence (that is, Rww' means that time w is earlier than time w'). Consider the (directed) graph in Figure 1. This shows a

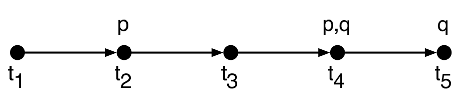


Figure 1. A simple temporal model.

simple flow of time consisting of five points. Here we will take the precedence relation to be the transitive closure of the next-time relation indicated by the arrows (after all, we think of the flow of time as transitive) thus every point t_i precedes all points to its right. Note that (as we would expect from the internal perspective provided by modal languages) whether or not a formula is satisfied depends on where (or in this example, *when*) it is evaluated. For example, the formula $\Diamond(p \wedge q)$ is satisfied at points t_1, t_2 and t_3 (because all these points are to the left of t_4 where both p and q are true together) but not at t_4 and t_5. On the other hand, because q is true at t_5, we have that $\Diamond q$ is true at t_1, t_2, t_3 and t_4. One special case is worth remarking on: note that for any basic formula φ whatsoever, $\Box\varphi$ is satisfied at t_5. Why? Because the clause in the satisfaction definition for boxes says that $\Box\varphi$ is satisfied if and only if φ is satisfied at *all* R-accessible points. As no points are R-accessible from t_5 (it has no points to its right) this condition is trivially met.

The idea of using modal logic as a tool for temporal reasoning is due to Arthur Prior [104, 105]. His work offers what is probably the clearest example of modal logic being appreciated for its internal perspective. In languages such as English and Dutch, the default way of locating information temporally is to use tenses, and tenses locate information *relative* to the point of speech. For example, if at some time t I say "Clarence will fly", then this will be true if at some future time t' Clarence does in fact fly. Prior viewed tensed talk as fundamental: we exist in time, and have to deal with temporal information from the inside. He believed that the internal perspective offered by modal languages made it an ideal tool for capturing the situated nature of our experience and the context-dependent way we talk about it. Prior called his system *tense logic*. He wrote F for the forward looking (or future) diamond, and had a second diamond, written P, for looking back into the past (so in Figure 1, $P(p \wedge q)$ is true at t_5, for this point is to the right of t_4, where p and q are true together). Prior needed backward looking operators to mimic the effect of natural language past tense constructions; for further discussion of Prior's work in this area, see Chapter 19 of this handbook.

Our next example brings us to one of the most influential ways of thinking about Kripke models: to give them a *process interpretation*, which means that we view models as collections of computational states, and the binary relations as computational actions that transform one state into another. This interpretation dates back to the classic work of Hoare [67] and Dijkstra [32]. Let's look at a simple example. Consider the graph shown in Figure 2. This shows a finite state

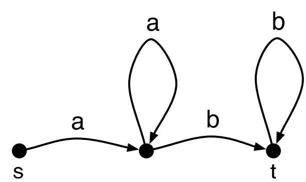


Figure 2. Finite state automaton for $a^n b^m (n, m > 0)$.

automaton for the formal language $a^n b^m$ $(n, m > 0)$, that is, for the set of all strings consisting of a non-empty block of as followed by a non-empty block of bs. But this is precisely the type of graph we can use to interpret a modal language. In this case it would be natural to work with a language with two diamonds $\langle a \rangle$ and $\langle b \rangle$. The $\langle a \rangle$ diamond will be used to explore the a-transitions in the automaton, while the $\langle b \rangle$ diamond explores the b-transitions. It follows that all formulas of the form

$$\langle a \rangle \cdots \langle a \rangle \langle b \rangle \cdots \langle b \rangle t$$

(that is, an unbroken block of $\langle a \rangle$ diamonds preceding an unbroken block of $\langle b \rangle$ diamonds in front of a proposition symbol t which is only true at the terminal node t) are satisfied at the start node s, for all modality sequences of this form correspond to the strings accepted by the automaton. Although simple, this example shows the key feature of many computational interpretations of modal logic: the relations are thought of as processes (here our processes are "read the symbol a" and "read the symbol b"). Note that in this case we are thinking in terms of deterministic processes (each relation is a partial function) but we could just as well work with arbitrary relations, which amounts to working with a non-deterministic model of processes. The process interpretation, in various forms, underlies much of the discussion of this chapter, and it underlies Chapters 12 and 17 of this handbook.

Another important application of modal languages is to model the logic of knowledge and

belief; this line of work was pioneered by Jaakko Hintikka [66], and as the more recent treatise by Fagin, Halpern, Moses, and Vardi [39] makes clear, the study of *epistemic logic* continues to flourish. Again, simple graph-based intuitions underly this application. Consider, for example, the graph shown in Figure 3. Here we see the epistemic state of a very simple agent. One

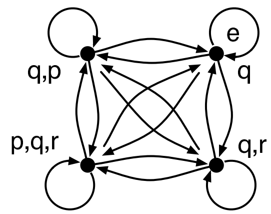


Figure 3. Epistemic state of a simple agent.

of the epistemic situations making up this state is marked e; this represents the agent's current knowledge (the agent knows that q is the case). The other situations represent the way the world might be. For example, although neither p nor r are true in the current situation, the agent views situations in which p and q are true together, and situations in which r and q are true together, and even situations in which p and q and r are all true together, as epistemically acceptable alternatives to the current situation e. So $\Diamond(p \wedge q)$ ("$p \wedge q$ is consistent with what the agent knows"), and $\Diamond(r \wedge q)$, and $\Diamond(p \wedge q \wedge r)$ are all satisfied at e. Moreover $\Box q$ ("the agent knows that q") is satisfied at e, as at every alternative epistemic situation the information q holds. Hintikka introduced the symbol K for this usage of box (that is, he wrote Kq for "the agent knows that q") and his notation is still standard in contemporary epistemic logic. Epistemic logic is discussed in Chapters 18 and 20 of this handbook.

The next example is important for another reason. Modal logic is often viewed as an intrinsically *intensional* logic, interpreted using *possible world semantics*. This view comes from what is probably the most historically influential interpretation of modal logic, namely as the logic of necessity and possibility. In this interpretation, $\Diamond$ is read as "possibly", $\Box$ is read as "necessarily", and the points of Kripke models are regarded as possible worlds. Unfortunately, this interpretation has tended to overshadow the others, at least in certain research communities (some philosophers view modal logic, intensionality, and possible worlds as inextricably intermingled). To ensure that this illusion is dispelled, our last example will be completely *extensional*. Consider the graph in Figure 4.

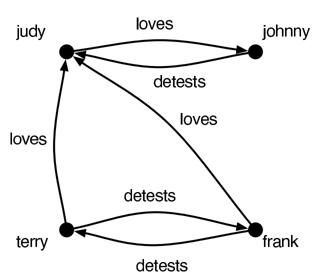


Figure 4. Ordinary individuals.

This is the sort of extensional information that classical logics (such as first-order logic) are

often used for. But modal logic is at home here too. We can say lots of interesting things about such situations. For example

$$\langle \text{LOVES} \rangle \top \wedge \langle \text{DETESTS} \rangle \langle \text{LOVES} \rangle \top$$

is true when evaluated at Terry: he loves someone and he detests someone who loves someone. Nowadays, modal logic is widely used for reasoning about such extensional situations. In particular, the concept languages which lie at the heart of the *description logics* used in knowledge representation are often notational variants of (various kinds of) modal languages. Description logics are used in a wide range of applications for representing and reasoning about extensional information. They are treated in depth in Chapter 13 of this handbook.

We're almost ready to define the standard translation, but before doing so let's deal with three other matters. First, in most branches of logic and mathematics, there is a notion of two structures being *isomorphic*, which can be glossed as "mathematically indistinguishable". Let's take this opportunity to be precise about what isomorphism means in basic modal logic (we give the definition for models and frames with one relation; it generalises straightforwardly to structures with multiple relations).

DEFINITION 1 (Isomorphism). Let $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$ be models, and $f : W \mapsto W'$ a bijection. If for all $w, v \in W$ we have that Rwv if and only if $Rf(w)f(v)$ then we say that f is an isomorphism between the frames (W, R) and (W', R') and that these frames are isomorphic. If in addition we have, for all proposition symbols p, that $w \in V(p)$ if and only if $f(w) \in V'(p)$ then we say that f is an isomorphism between the models $\mathfrak{M}$ and $\mathfrak{M}'$ and that these models are isomorphic.

As this definition makes clear, if models $\mathfrak{M}$ and $\mathfrak{M}'$ are isomorphic, each replicates perfectly the information in the other. Hence the following result is unsurprising:

PROPOSITION 2. *Let f be an isomorphism between models $\mathfrak{M}$ and $\mathfrak{M}'$. Then for all basic modal formulas φ, and all points w in $\mathfrak{M}$, we have that $\mathfrak{M}, w \models \varphi$ if and only if $\mathfrak{M}, f(w) \models \varphi$.*

Proof. Immediate by induction on the construction of φ (see Lemma 9 for an example of such a proof.) ❑

Second, we want to point out that it is possible to take a more dynamic perspective on the satisfaction definition. In particular, we can think of it as a game. Let's start with a concrete example. Consider the model in Figure 5.

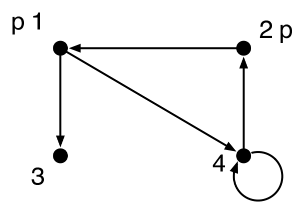


Figure 5. The formula $\Diamond\Box\Diamond p$ is true at 1 and 4, but false at 2 and 3.

As the reader should check, $\Diamond\Box\Diamond p$ is true at points 1 and 4, but false at points 2 and 3. Now suppose we play the following *evaluation game*. This game has two players, a Verifier (V) and a

Falsifier (F), who disagree about the satisfiability of a formula in some model. The two players react differently to the connectives in the formula: for example, occurrences of disjunction allow V to make a choice as to which disjunct to verify, and play continues with the formula chosen; negation switches the roles of the two players; and diamonds make V pick a successor of the current point, while boxes do the same for F. Moreover, for any proposition symbol p, V wins the p-game if p is true at the current state, otherwise F wins. A player also wins the game if the other player must make a move for a modality but cannot.

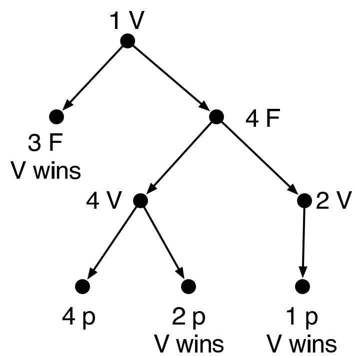


Figure 6. Initial segment of a game tree.

So let's play the game for $\Diamond\Box\Diamond p$ at 1. Figure 6 shows (an initial segment of) the resulting game tree. Note that V can always win. Her most obvious option is to play 3 in response to the outermost diamond; this leaves F with no possible response when faced with the task of falsifying $\Box\Diamond p$. But V can also safely play 4 on her first move. As the tree shows, irrespective of F's response, V can always reach a winning position. What this example suggests is completely general: for any model $\mathfrak{M}$, point w, and basic formula φ, we have that $\mathfrak{M}, w \models \varphi$ if and only if V has a winning strategy when the φ-game is played in $\mathfrak{M}$ starting at w. Moreover, as we see in this example, different strategies correspond to different ways of showing that the given formula is true.

Finally, some historical remarks. Where does the relational interpretation of modal logic come from? The three authors usually cited as pioneers are Saul Kripke, Jaakko Hintikka, and Stig Kanger. Kripke's contributions are the best known (indeed relational semantics is often called Kripke semantics) and Kripke [83, 84] are regarded as landmarks in the development of modal semantics. But Hintikka independently developed the idea in his work on logics of knowledge and belief (see, for example, his classic monograph "Knowledge and Belief" [66]). Furthermore, although his work was not well known at the time, Kanger, in a series of papers and monographs published in 1957, introduced relational semantics for modal logic (see, for example, Kanger [77, 78]). Indeed, the idea of relational semantics seems to have been in the air at around this time, and a number of other logicians (for example Arthur Prior and Richard Montague) discussed similar ideas. For a detailed discussion of who did what and when, the reader should consult Goldblatt [59].

2.2 The standard translation

We now understand what modal languages are, how they can be interpreted in graphs, and why this can be an interesting thing to do. What next? Well, if we were following a traditional path, we would probably remark that as modal languages are to be used for reasoning, some sort of proof system is called for. For example, if we were working in a language with one modality (and in which we had chosen to define $\Diamond$ in terms of $\Box$) we might point out that the set of all modal validities (that is, the *minimal modal logic*) in the language could be axiomatised by a Hilbert-style proof system called **K**. This proof system can be defined in a number ways; we might, for example, stipulate that the axioms of **K** consist of all formulas in the language which have the form of a propositional tautology (by which we mean not merely tautologies such as $p \to p$ which contain no modalities, but also formulas such as $\Box\Diamond p \to \Box\Diamond p$, which contain modalities but are truth-functionally tautologous too) and all instances of the following axiom schema:

$$\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi).$$

There are two rules of proof: *modus ponens* (if $\vdash \varphi$ and $\vdash \varphi \to \psi$ then $\vdash \psi$) and *modal generalisation* (if $\vdash \varphi$ then $\vdash \Box\varphi$); in the definitions of these rules, $\vdash \theta$ is standard notation that means "the formula θ is provable". Now, this looks like a standard axiomatisation of first-order logic with $\Box$ behaving like $\forall$. But **K** has no analogs of the first-order axioms with tricky side conditions on freedom and bondage of variables, such as $\forall x\varphi \to [\tau/x]\varphi$, where τ is a first-order term. This is no coincidence. As the standard translation given below will make clear, modal logic is essentially a perspicuous variable-free notation for a fragment of first-order logic.

But proof systems are not our goal. This chapter is concerned with semantic issues, so quite different aspects of modal logic call for our attention. To get the ball rolling, let's return to our basic semantic entities (Kripke models) and ask what they actually are. This will provide a point of entry to one of the main themes of the chapter: the relationship between modal and classical logic.

So, what is a Kripke model? No mystery here. A Kripke model $(W, \{R^m\}_{m \in \text{MOD}}, V)$ is what model theorists call a *relational structure*. That is, we have a domain of quantification W, a collection of binary relations over this domain, and a collection of unary relations as well (after all, $V(p)$ is a unary relation for each $p \in$ PROP). But this means that we are not forced to talk about Kripke models using modal languages: they provide us with everything needed to interpret classical languages too. For example, to talk about a model $(W, \{R^m\}_{m \in \text{MOD}}, V)$ using first-order logic we would simply make use of a first-order language with a binary relation symbol R^m for every $m \in$ MOD, and a unary relation symbol P for every $p \in$ PROP. Modal logicians have a name for this language: they call it the *first-order correspondence language* (for the basic modal language over PROP and MOD).

Why "correspondence language"? Because every basic modal formula (in the language over PROP and MOD) corresponds to a first-order formula from this language via the *standard translation*:

$$\begin{array}{rcl}
\text{ST}_x(p) & = & Px \\
\text{ST}_x(\bot) & = & \bot \\
\text{ST}_x(\neg\varphi) & = & \neg\, \text{ST}_x(\varphi) \\
\text{ST}_x(\varphi \wedge \psi) & = & \text{ST}_x(\varphi) \wedge \text{ST}_x(\psi) \\
\text{ST}_x(\langle m \rangle\varphi) & = & \exists y(R^m xy \wedge \text{ST}_y(\varphi)) \\
\text{ST}_x([m]\varphi) & = & \forall y(R^m xy \to \text{ST}_y(\varphi)).
\end{array}$$

That is, the standard translation maps proposition symbols to unary predicates, commutes with booleans, and handles boxes and diamonds by explicit first-order quantification over R^m-accessible points. The variable y used in the clauses for diamonds and boxes is chosen to be any new variable (that is, one that has not been used so far in the translation). We remarked earlier that diamonds and boxes were essentially a simple macro notation encoding quantification over accessible states; the standard translation expands these macros. Note that $\mathrm{ST}_x(\varphi)$ always contains exactly one free variable (namely x). This free variable is what allows the internal perspective, typical of modal logic, to be mirrored in a classical language: assigning a value to this variable is analogous to evaluating a modal formula inside a model at a certain point.

Here's an example of the translation at work:

$$\begin{array}{rcl} \mathrm{ST}_x(p \to \Diamond p) & = & \mathrm{ST}_x(p) \to \mathrm{ST}_x(\Diamond p) \\ & = & Px \to \mathrm{ST}_x(\Diamond p) \\ & = & Px \to \exists y(Rxy \wedge \mathrm{ST}_y(p)) \\ & = & Px \to \exists y(Rxy \wedge Py). \end{array}$$

As the reader can easily check, $p \to \Diamond p$ and its standard translation $Px \to \exists y(Rxy \wedge Py)$ are equisatisfiable in the following sense: for any model $\mathfrak{M}$, and any point w in $\mathfrak{M}$, we have that $\mathfrak{M}, w \models p \to \Diamond p$ if and only if $\mathfrak{M} \models Px \to \exists y(Rxy \wedge Py)[x \leftarrow w]$, where the notation $[x \leftarrow w]$ means assign w to the free variable x. Unsurprisingly, this relationship is completely general:

PROPOSITION 3. *For any basic modal formula φ, any model $\mathfrak{M}$, and any point w in $\mathfrak{M}$, we have that $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M} \models \mathrm{ST}_x(\varphi)[x \leftarrow w]$.*

Proof. There is practically nothing to prove. The clauses of the standard translation mirror the clauses of the satisfaction definition. Hence the result is immediate by induction on the structure of modal formulas. ❑

Thus the standard translation gives us a bridge between modal logic and classical logic. And we can immediately use this bridge to transfer meta-theoretic results for first-order logic to modal logic.

PROPOSITION 4. *Basic modal logic has the compactness property. That is, if Σ is a set of basic modal formulas, and every finite subset of Σ is satisfiable, then Σ itself is satisfiable. Moreover, basic model logic has the Löwenheim-Skolem property. That is, if a set of basic modal formulas Σ is satisfiable in at least one infinite model, then it is satisfiable in models of every infinite cardinality.*

Proof. We show that basic modal logic has the Löwenheim-Skolem property. Suppose that Σ is a set of basic modal formulas that has at least one infinite model. Let $\mathrm{ST}_x(\Sigma)$ be the set of (first-order) formulas obtained by standardly translating all the formulas in Σ. Now, as Σ has an infinite model, by Proposition 3 so does $\mathrm{ST}_x(\Sigma)$. But first-order logic has the Löwenheim-Skolem property, hence $\mathrm{ST}_x(\Sigma)$ has a model of every infinite cardinality. But, again by appeal to Proposition 3, each of these models satisfies Σ, so basic modal logic has the Löwenheim-Skolem property too. The argument showing it has the compactness property is similar. ❑

Another easy consequence of the standard translation is that the set of validities (in basic modal languages) is recursively enumerable. For a basic modal formula φ is valid iff $\mathrm{ST}_x(\varphi)$ is a first-order validity, and the set of first-order validities is recursively enumerable.

Let's sum up what we have learned so far. Propositional modal languages are syntactically simple languages that offer a neat (variable-free) notation for talking about relational structures. They talk about relational structures from the inside, using the modal operators to look for information at accessible states. This internal perspective on models, coupled with the simplicity of modal syntax, means that propositional modal logic is an attractive tool for certain applications. Moreover, viewed as a tool for talking about models, any basic modal language can be regarded as a fragment of its corresponding first-order language: the standard translation systematically maps modal formulas to first-order formulas (in one free variable) and makes the quantification over accessible states explicit. This allows us to quickly establish some basic modal meta-theory by appeal to known results for first-order logic.

3 BISIMULATION AND DEFINABILITY

With the basics behind us it is time to look deeper. In particular, it is time to start mapping the expressive strengths and weaknesses of the basic modal language. Now, the expressive power of a language is usually measured in terms of the distinctions it can draw. A language with just the two expressions "like" and "dislike" would provide only the roughest possible classification of the world, whereas a richer language of assent and dissent would make it possible to draw finer distinctions inside the accepted and rejected situations. So what distinctions can modal languages draw? In this section we discuss this question at the level of models, and in Section 5 we shall reconsider it at the level of frames. In what follows it will often be useful to think in terms of *pointed models*. That is, we shall often present models together with an explicit distinguished point to indicate where we are trying to find a difference.

3.1 Drawing distinctions

A modal language (and indeed any logical language whose formulas form a set) can distinguish between some models $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$, but not between all such pairs. For example, our basic modal language can distinguish the pair of models shown in Figure 7.

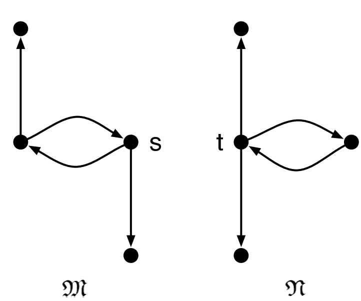


Figure 7. $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$ are modally distinguishable.

Here $\Box(\Box\bot \vee \Diamond\Box\bot)$ is a modal formula that distinguishes these models: it is true in $\mathfrak{M}$ at s, but false in $\mathfrak{N}$ at t. But now consider the pair of models shown in Figure 8. Is it possible to *modally* distinguish $(\mathfrak{M}, s)$ from $(\mathfrak{K}, u)$? That is, is it possible to find a (basic) modal formula that is true in $\mathfrak{M}$ at s, but false in $\mathfrak{K}$ at u? Note that it is easy to distinguish them if we are

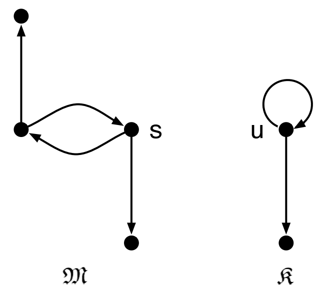


Figure 8. $(\mathfrak{M}, s)$ and $(\mathfrak{K}, u)$ are not modally distinguishable.

allowed to use first-order logic: all points in $\mathfrak{M}$ (including s) are irreflexive, while point u in $\mathfrak{K}$ is reflexive, hence the first-order formula Rxx is not satisfiable (under any variable assignment) in model $\mathfrak{M}$, but it is satisfied in $\mathfrak{K}$ when u is assigned to x. But no matter how ingenious you are, you will not find any formula in the basic modal language that distinguishes these models at their designated points. Why is this?

3.2 Bisimulation

A natural approach to this question is to consider its dual: when should two models be viewed as modally identical? For example, given a process interpretation, when would we view two transition diagrams as representations of the same process? The models $\mathfrak{M}$ and $\mathfrak{K}$ of Figure 8 provide an intuitive example: they seem to stand for the same process when we look at possible actions and deadlocks (note that at each state the process can enter a deadlock situation; that is, it can enter a state from which it cannot exit). By contrast, $\mathfrak{M}$ and $\mathfrak{N}$ in Figure 7 are different, as the right hand state in $\mathfrak{N}$ is not threatened with immediate dead-lock. Or consider the epistemic interpretation: when would we want to say that two graphs represent the same epistemic state? For example, we would probably want to identify the two epistemic models shown in Figure 9 at their distinguished points s and t.

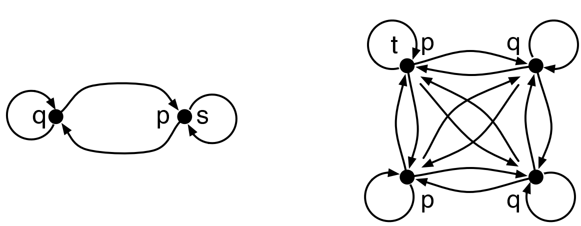


Figure 9. Two epistemically equivalent models.

After all, in essence both models present us with a two way choice: either we are in an epistemic situation where p holds and there is an accessible epistemic situation where q holds, or we are in an epistemic situation where q holds and there is an accessible epistemic situation where p holds. The intuition that both these graphs code the same epistemic state is captured by our

modal language: the reader will not find any modal formula that distinguishes them.

The modal logician's idea of asking when two distinct structures are modally identical (that is, make the same modal formulas true) lies within an older (and broader) tradition of looking for the structure preserving morphisms in a given mathematical domain, and letting the corresponding theory describe those notions that are invariant for such morphisms. This is the spirit of Klein's Program in geometry, proposed around 1870, and still influential in many fields. Of course, there is no unique answer to the question of when two structures are the same. This insight was stated forcefully in recent years by President Clinton during the Lewinsky hearings: *It all depends on what you mean by "is"*. Clinton's Principle for modal logic means that we should first try to stipulate some notion of structural equivalence for models that is appropriate for modal languages. This is the purpose of the following definition (first formulated in van Benthem [128, 131]). We state it here for models with one relation R, but the definition generalises straightforwardly to models with any number of relations.

DEFINITION 5 (Bisimulation). A bisimulation between models $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$ is a non-empty binary relation E between their domains (that is, $E \subseteq W \times W'$) such that whenever wEw' we have that:

Atomic harmony: w and w' satisfy the same proposition symbols,

Zig: if Rwv, then there exists a point v' (in $\mathfrak{M}'$) such that vEv' and $R'w'v'$, and

Zag: if $R'w'v'$, then there exists a point v (in $\mathfrak{M}$) such that vEv' and Rwv.

If there is a bisimulation between two models $\mathfrak{M}$ and $\mathfrak{N}$, then we say that $\mathfrak{M}$ and $\mathfrak{N}$ are bisimilar. Moreover, we say that two states are bisimilar if they are related by some bisimulation.

Putting this in words: two states are bisimilar if they make the same atomic information true and if, in addition, their transition possibilities match. That is, if a transition to a related state is possible in one model, then the bisimulation must deliver a matching transition possibility in the other. Atomic harmony, coupled with the matching transitions concept embodied in the zigzag clauses, make bisimulation a natural notion of process equivalence, and indeed bisimulations were independently discovered in computer science (see Park [100]).

Returning to the models $\mathfrak{M}$, $\mathfrak{K}$, and $\mathfrak{N}$ considered above (and disregarding proposition symbols) it is easy to see that $\mathfrak{M}$ and $\mathfrak{K}$ are bisimilar: the dotted lines in Figure 10 indicate the required bisimulation (note that the indicated bisimulation links the two designated points). Furthermore, it is easy to see that there is no bisimulation that links the designated points of $\mathfrak{N}$ and $\mathfrak{K}$. Why not? Because a move from t to the right-hand world in $\mathfrak{N}$ has no matching move in $\mathfrak{K}$: moving downwards from u is no option (end-points never bisimulate with points having successors) but neither is moving reflexively from u to itself (as one can move from u to a successor which is an endpoint, but this can't be done from the right-hand world in $\mathfrak{N}$).

Given any modal model $\mathfrak{M}$, bisimulations can be used in a number of ways. The so-called *bisimulation contraction* makes $\mathfrak{M}$ as small as possible. To define this, note that it follows from Definition 5 that any union of bisimulations between two models is itself a bisimulation. Therefore the union of all bisimulations between two models is a maximal bisimulation between them. Now form the maximal bisimulation of model $\mathfrak{M}$ with itself (incidentally, a bisimulation of a model with itself is called an *autobisimulation*). Define a quotient of $\mathfrak{M}$ whose points are the equivalence classes, and relate the equivalence class $|w|$ to the equivalence class $|v|$ iff $|w|$ and $|v|$ contain points w' and v' such that $Rw'v'$. The map from points to their equivalence classes is a bisimulation. For example, the bisimulation shown in Figure 10 between $\mathfrak{M}$ and $\mathfrak{K}$ is a bisimulation contraction. Bisimulation contractions are the most compact representation of processes,

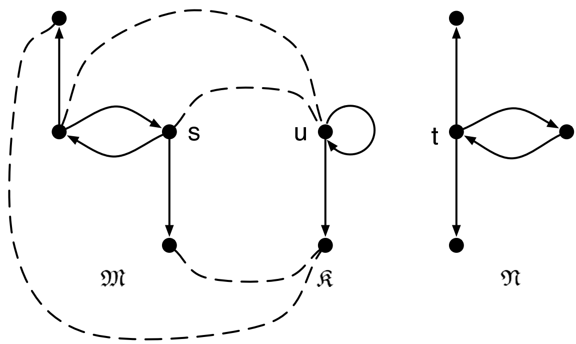


Figure 10. $(\mathfrak{M}, s)$ and $(\mathfrak{K}, u)$ are bisimilar, $(\mathfrak{K}, u)$ and $(\mathfrak{N}, t)$ are not.

at least from a modal standpoint. They remove all the redundancies in the representation — but also all aesthetic symmetries. (A butterfly is a redundant object, as one wing contains enough information under this perspective.)

Bisimulations can also be used to make bigger models: one important construction which does this is called *tree unraveling* (for a very early paper using this construction, see Dummett and Lemmon [34]; for an influential paper that made heavy use of it, see Sahlqvist [111]).

To unravel a model, take all finite R-sequences of points in $\mathfrak{M}$ that start at some point w. These sequences form a tree with one-step extensions of sequences as the tree-successor relation. Projection from a sequence to its last element is a bisimulation onto the original model $\mathfrak{M}$. As an example, consider the unraveling of the two element model $\mathfrak{K}$ around its distinguished point u to the infinite comb-like structure shown in Figure 11 (we use v as the name of the other point in this model). Reasoning about trees is often easier than reasoning about arbitrary graphs, and

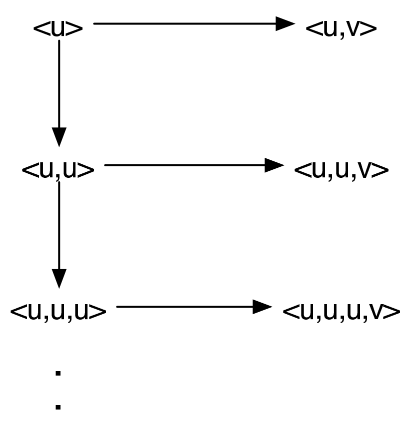


Figure 11. Unraveling $\mathfrak{K}$ around u.

so this method is of considerable theoretical utility. Moreover, as we shall see in the following

section, tree unraveling is relevant to the *decidability* of modal logic.

Three other model constructions used in modal logic, namely *disjoint unions*, *generated submodels*, and *bounded morphisms* (or *p-morphisms*) are also bisimulations. Historically, all three constructions were widely used in modal logic more than a decade before the unifying concept of bisimulation was introduced (the classic source for these constructions is Segerberg [113], where they are heavily used, often in combination, to prove completeness theorems). All three constructions are fundamental tools in many areas of modal logic (for example, when reformulated at the level of frames, they are key ingredients in the Goldblatt-Thomason Theorem which we discuss in Section 5) so we take this opportunity to define them for models with one accessibility relation. These definitions generalise straightforwardly to models of arbitrary signature.

The simplest construction is forming disjoint unions. If we have a pair of disjoint models (that is, a pair of models (W, R, V) and (W', R', V') such that W and W' are disjoint) then their disjoint union is the model $(W \cup W', R \cup R', V + V')$, where $V + V'$ is the valuation defined by $(V + V')(p) = V(p) \cup V'(p)$, for all proposition symbols p. That is, forming a disjoint union of two models means lumping together all the information in the two graphs. What if the graphs are not disjoint? Then we simply take disjoint isomorphic copies of the two models, and form the disjoint union of the copies. This lumping together process can be generalised to arbitrarily many models, which prompts the following definition.

DEFINITION 6 (Disjoint Unions). Given mutually disjoint models $\mathfrak{M}_i = (W_i, R_i, V_i)$, where i ranges over the elements of some index set I, we define the disjoint union of these models to be $\mathfrak{M} = (W, R, V)$, where $W = \bigcup_{i \in I} W_i$, $R = \bigcup_{i \in I} R_i$, and $V(p) = \bigcup_{i \in I} V_i(p)$ for all proposition symbols p. To form the disjoint union of a collection of models that are not mutually disjoint, we first take mutually disjoint isomorphic copies, and then form the disjoint union of the copies.

It is immediate from this definition that any component model $\mathfrak{M}_i$ of a disjoint union $\mathfrak{M}$ is bisimilar with $\mathfrak{M}$: for the bisimulation relation E we simply take the identity relation. Identity clearly satisfies the atomic harmony and zigzag conditions required of bisimulations.

Disjoint unions build bigger models from (collections of) smaller ones. Generated submodels do the reverse. They arise by restricting attention to subgraphs of a given graph that are closed under relational transitions. For example, consider the two graphs in Figure 12. It is clear that

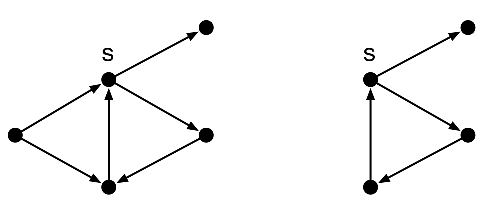


Figure 12. Generating a submodel from s.

the graph on the right arises by restricting attention to a certain transition-closed subgraph of the graph on the left, namely the set of point reachable by taking sequences of transitions from s. This motivates the following definition.

DEFINITION 7 (Generated Submodels). Let $\mathfrak{M} = (W, R, V)$ be a model and let $W' \subseteq W$. We say that a model $\mathfrak{M}' = (W', R', V')$ is the restriction of $\mathfrak{M}$ to W' if $R' = R \cap (W' \times W')$ and for all proposition symbols p we have that $V'(p) = V(p) \cap W'$. We say that W' is R-closed

if for all $u \in W'$, if Ruv then $v \in W'$. Finally, we say that $\mathfrak{M}'$ is a generated submodel of $\mathfrak{M}$ iff $\mathfrak{M}'$ is the restriction of $\mathfrak{M}$ to an R-closed subset of W.

If $\mathfrak{M}' = (W', R', V')$ is a generated submodel of $\mathfrak{M} = (W, R, V)$, and $S \subseteq W'$ has the property that every $w' \in W'$ is reachable via a finite sequence of R-transitions from some $s \in S$, then we say that $\mathfrak{M}'$ is the submodel of $\mathfrak{M}$ generated by S. If S is a singleton set $\{s\}$, then we say that $\mathfrak{M}'$ is the submodel of $\mathfrak{M}$ generated by the point s.

A generated submodel is bisimilar to the model that gave rise to it: as with disjoint unions, the identity relation relates the two models in the appropriate way. Incidentally, note that every component model of a disjoint union is a generated submodel of the disjoint union.

Finally we turn to bounded morphisms (or p-morphisms as they are often called).

DEFINITION 8 (Bounded Morphisms). A bounded morphism between models $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$ is a function f with domain W and range W' such that:

Atomic harmony: Points in W and their f-images satisfy the same proposition symbols (that is, $w \in V(p)$ iff $f(w) \in V'(p)$, for all proposition symbols p).

Morphism: if Rwv, then $R'f(w)f(v)$.

Zag: if $R'f(w)v'$, then there exists a v (in $\mathfrak{M}$) such that $f(v) = v'$ and Rwv.

If f is a bounded morphism from $\mathfrak{M}$ to $\mathfrak{M}'$ and f is surjective, then we say that $\mathfrak{M}'$ is a bounded morphic image of $\mathfrak{M}$.

Bounded morphisms are bisimulations: a bounded morphism is simply a bisimulation in which the bisimulation relation E is an R-preserving morphism f (note that the only essential difference between the two definitions is that the morphism clause replaces the zig clause, and clearly morphism implies zig). Historically, it was the definition of bounded morphisms that inspired the definition of bisimulations.

As an example of a bounded morphism between models, consider Figure 13 (again we ignore proposition symbols).

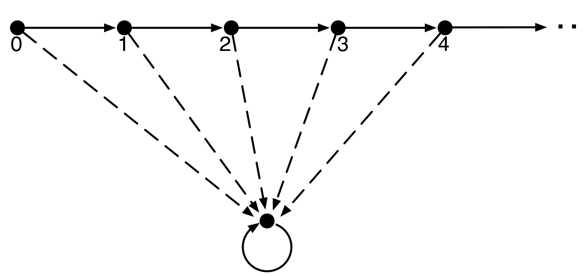


Figure 13. Bounded morphism collapsing the natural numbers to a reflexive point.

Here we have collapsed the natural numbers in their usual order to a single reflexive point. It is clear that this map satisfies both the morphism and zig clauses, so it is indeed a bounded morphism.

3.3 Invariance and definability in first-order logic

Structural invariances preserve certain patterns definable in appropriate languages. Before pursuing the match between bisimulation and modal logic, let us examine the situation in first-order logic. The archetypal structural invariance is *isomorphism* between models. As we saw earlier (recall Proposition 2) modal formulas are invariant for isomorphism. More generally, it is well known that if f is an isomorphism between $\mathfrak{M}$ and $\mathfrak{N}$, then for each first-order formula $\varphi(x_1, \ldots, x_k)$, and each matching tuple of objects $\langle d_1, \ldots, d_k \rangle$ in $\mathfrak{M}$, the following equivalence holds:

$$\mathfrak{M} \models \varphi[d_1, \ldots, d_k] \text{ iff } \mathfrak{N} \models \varphi[f(d_1), \ldots, f(d_k)],$$

or stated in words: first-order formulas are invariant for isomorphism.

On special models, the converse also holds. For example, it is a well-known fact that any two finite models with the same first-order theory are isomorphic. But no general converse holds, as there are many more isomorphism classes of models than complete first-order theories. Invariance for isomorphism is even a defining condition for any logic in abstract model theory. But no matter how strong the logic, the converse still fails whenever the formulas of a logic form a set, as opposed to the proper class of isomorphism types.

Thus it makes sense to look at invariance conditions for weaker notions of structural equivalence. For example, a *potential isomorphism* between two models $\mathfrak{M}$ and $\mathfrak{N}$ is a non-empty set I of finite partial isomorphisms satisfying the back-and-forth extension conditions that, whenever $f \in I$ and $d \in \mathfrak{M}$, then there is an $e \in \mathfrak{N}$ such that $f \cup \{(d, e)\} \in I$, and vice-versa. Note that isomorphisms induce potential isomorphisms: simply take I to be the family of all finite restrictions. The converse is not true. Matching up all finite sequences of rational numbers with equally long sequences of real numbers (in the same order) is a potential isomorphism between $\mathbb{Q}$ and $\mathbb{R}$, even though these two structures are not order-isomorphic for cardinality reasons.

It is easy to show that all first-order formulas are invariant for potential isomorphism, but the real match is with a stronger language: two models are potentially isomorphic iff they have the same complete theory in the *infinitary* first-order logic $\mathcal{L}_{\infty\omega}$. This formalism also gives rise to much stronger definability results. For example, for each model $\mathfrak{M}$ there is a sentence $\delta_{\mathfrak{M}}$ of $\mathcal{L}_{\infty\omega}$ which holds only in those models $\mathfrak{N}$ which have a potential isomorphism with $\mathfrak{M}$; that is, models can be defined up to potential isomorphism. Moreover, countable models can even be defined (modulo isomorphism) using only countable conjunctions and disjunctions. This is all very nice of course, but infinitary logic is a bit outlandish from a practical viewpoint.

Better matches between structural invariance and first-order definability arise in the more fine-grained setting of Ehrenfeucht-Fraïssé comparison games between models $\mathfrak{M}$ and $\mathfrak{N}$ played between a Spoiler (who looks for differences between the models) and a Duplicator (who looks for analogies between them). Models $\mathfrak{M}$ and $\mathfrak{N}$ have the same first-order theory up to quantifier depth k iff the Duplicator has a winning strategy in their comparison game over k rounds. We won't give details here, as we will define a modal comparison game of this sort at the end of the section.

3.4 Invariance and definability in modal logic

With these analogies in mind, let us now investigate the modal situation. For a start, modal formulas are *invariant for bisimulation*:

LEMMA 9 (Bisimulation Invariance Lemma). *If E is a bisimulation between $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$, and wEw', then w and w' satisfy the same basic modal formulas.*

Proof. By induction on the construction of modal formulas. The case for proposition symbols is immediate by atomic harmony. The inductive steps for the boolean connectives are straightforward. And the inductive step for $\Diamond$ formulas shows exactly what the zigzag clauses were designed for. For consider the left to right direction. Given $\mathfrak{M}, w \models \Diamond\varphi$ and wEw', we want to show that $\mathfrak{M}', w' \models \Diamond\varphi$. Now, $\mathfrak{M}, w \models \Diamond\varphi$ means that there is some v in $\mathfrak{M}$ such that Rwv and $\mathfrak{M}, v \models \varphi$. But then (by zig) there must be a point v' in $\mathfrak{N}'$ such that vEv' and $R'w'v'$. By the induction hypothesis, $\mathfrak{M}', v' \models \varphi$, hence $\mathfrak{M}', w' \models \Diamond\varphi$ as required. The argument for the right to left direction is essentially the same, using zag in place of zig. ❑

The result allows us to show failures of bisimulation easily. For example, we have already sketched an argument showing that the models $\mathfrak{N}$ and $\mathfrak{K}$ of Figure 10 have no bisimulation between their designated points, but a quicker proof is now possible: these points *cannot* be bisimilar because there are modal formulas (for example $\Box(\Box\bot \vee \Diamond\Box\bot)$) which are satisfied at one point but not the other. On the other hand, the dotted lines in Figure 10 show that $\mathfrak{M}$ and $\mathfrak{K}$ are bisimilar; it follows that all points linked by a dotted line in these graphs make exactly the same modal formulas true. Another typical application of this result is to show the undefinability of certain structural notions. For example, we can show that irreflexivity is modally undefinable: no modal formula holds in exactly those points w of models such that $\neg Rww$. To prove this, it suffices to find two bisimilar points in two models, one of which is reflexive, the other irreflexive. One such example is the bisimulation between the designated points of $\mathfrak{M}$ and $\mathfrak{K}$ shown in Figure 10. Another is the bounded morphism of Figure 13 which collapses the natural numbers to a single reflexive point.

Another consequence of this result is that the disjoint union, generated submodel, and bounded morphism constructions are all satisfaction preserving. More precisely:

LEMMA 10. *Modal satisfaction is invariant under the formation of disjoint unions, generated submodels, and bounded morphisms. That is:*

1. *If $\mathfrak{M} = (W, R, V)$ is the disjoint union of $\mathfrak{M}_i = (W_i, R_i, V_i)$, for i from some index set I, then for all $w \in W_i$ and all $i \in I$ we have that $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M}_i, w \models \varphi$.*

2. *If $\mathfrak{M}' = (W', R', V')$ is a generated submodel of $\mathfrak{M} = (W, R, V)$, then for all $w' \in W'$ we have that $\mathfrak{M}, w' \models \varphi$ iff $\mathfrak{M}', w' \models \varphi$.*

3. *If $\mathfrak{M}' = (W', R', V')$ is a bounded morphic image of $\mathfrak{M} = (W, R, V)$ under the bounded morphism f, then for all $w \in W$ we have that $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M}', f(w) \models \varphi$.*

Proof. All three results could be proved by induction on the structure on φ. But such proofs are unnecessary: we know that disjoint unions, generated submodels, and bounded morphisms are all examples of bisimulations, hence these results follow from Lemma 9. ❑

To sum up the discussion so far, bisimulation implies modal equivalence. But what about the converse? For finite models, we have the following.

PROPOSITION 11. *If points w and w' from two finite models $\mathfrak{M}$ and $\mathfrak{N}$ satisfy the same modal formulas, then there is a bisimulation E between $\mathfrak{M}$ and $\mathfrak{N}$ such that wEw'.*

Proof. Assume we are working with models containing only a single relation R. We will show that the relation of modal equivalence is itself a bisimulation. That is, we will define the

bisimulation relation E by wEw' iff w and w' make the same modal formulas true. We now verify that E so defined is indeed a bisimulation.

It is immediate that E satisfies atomic harmony. As for zig, assume that wEw' and Rwv. Assume for the sake of contradiction that there is no v' in $\mathfrak{M}'$ such that $R'w'v'$ and vEv'. Let $S' = \{u' \mid R'w'u'\}$. Now, as w has an R-successor v, we have $\mathfrak{M}, w \models \Diamond\top$. As wEw', we have $\mathfrak{M}', w' \models \Diamond\top$ too, hence S' is non-empty. Furthermore, as $\mathfrak{M}'$ is finite, S' must be finite too, so we can write it as $\{u'_1, \ldots, u'_n\}$. By assumption, for every $u'_i \in S'$ there exists a formula ψ_i such that $\mathfrak{M}, v \models \psi_i$ but $\mathfrak{M}', u'_i \not\models \psi_i$. It follows that

$$\mathfrak{M}, w \models \Diamond(\psi_1 \wedge \cdots \wedge \psi_n) \text{ and } \mathfrak{M}', w' \not\models \Diamond(\psi_1 \wedge \cdots \wedge \psi_n),$$

which contradicts our assumption that wEw'. Hence E satisfies zig. A symmetric argument shows that E satisfies zag too, hence it is a bisimulation. ❑

Thus, on finite models, the expressive power of modal languages matches up exactly with bisimulation invariance. This result can be extended to broader model classes, such as models with finite branching width for successors (note that the proof just given does not depend on the models involved being finite: it would also work for infinite models in which each point has only finitely many R-successors) and suitably saturated models in a model-theoretic sense. But no general converse can hold, for the set-theoretic reasons mentioned earlier. Indeed, the converse does not hold generally even for countable models: not all modally equivalent countable models are bisimilar. Consider the two models in Figure 14 (assume that all proposition symbols are true at all points in both models). Both models have infinitely many branches leading away from their root nodes, but whereas all the branches in the model on the left are of finite length, the model on the right has a branch of infinite length. Now, as the reader should check, both models satisfy the same modal formulas at their root nodes. However there is no bisimulation that links their root nodes; the infinite branch in the model on the right makes it impossible to define one.

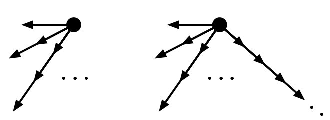


Figure 14. Modally equivalent but not bisimilar.

This counterexample could be repaired by passing to an *infinitary modal language* $\mathcal{L}^{\Diamond}_{\infty\omega}$ with arbitrary (countable) conjunctions and disjunctions. Infinitary modal equivalence occurs between countable models $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$ whenever there is a bisimulation linking s to t. Furthermore, every countable model $(\mathfrak{M}, s)$ is defined up to bisimulation by some $\mathcal{L}^{\Diamond}_{\infty\omega}$ formula $\delta_{\mathfrak{M},s}$. Again, such infinitary languages are somewhat impractical, but there are some useful bisimulation invariant formalisms which lie between the basic modal language and its infinitary extension. Two examples are *propositional dynamic logic* and the *modal μ-calculus*, which are discussed in Section 6.

Lemma 9 and its partial converses do not exhaust what needs to be said about the role played by bisimulations in modal model theory. But to gain a deeper understanding, we need to bring in a third component: the first-order correspondence language that we met in Section 2.2 when we introduced the standard translation.

3.5 Modal logic and first-order logic compared

The basic modal language can be viewed as a sort of miniature version of full first-order logic over graph models. The standard translation defined in the previous section shows that each modal formula φ corresponds to a first-order formulas $\mathrm{ST}_x(\varphi)$ containing a free variable x. But the converse does not hold: some first-order formulas in the correspondence language are not modally definable. We have already see an example. As the bisimulation between models $\mathfrak{M}$ and $\mathfrak{K}$ shows (recall Figure 10) no modal formula defines $\neg Rxx$. Thus, viewed as a tool for talking about models, modal logic is strictly less expressive than the full first-order correspondence language. And this prompts a further question: given that a modal language is essentially a fragment of the corresponding first-order language, exactly which fragment is it? This question has an elegant answer. First, a preliminary definition.

DEFINITION 12. A first-order formula $\varphi(x)$ is invariant for bisimulation if for all models $\mathfrak{M}$ and $\mathfrak{M}'$, and all points w in $\mathfrak{M}$ and w' in $\mathfrak{M}'$, and all bisimulations E between $\mathfrak{M}$ and $\mathfrak{M}'$ such that wEw', we have that $\mathfrak{M} \models \varphi[x \leftarrow w]$ iff $\mathfrak{M}' \models \varphi[x \leftarrow w']$.

We can now state the main result: basic modal languages correspond to the fragment of their first-order correspondence language that is invariant for bisimulation. More precisely:

THEOREM 13 (Modal Characterisation Theorem). *The following are equivalent for all first-order formulas $\varphi(x)$ in one free variable x:*

1. *$\varphi(x)$ is invariant for bisimulation.*
2. *$\varphi(x)$ is equivalent to the standard translation of a basic modal formula.*

Proof. That clause 2 implies 1 is a more or less immediate consequence of Lemma 9. The hard direction is showing that clause 1 implies 2. The original proof can be found in van Benthem [128, 131]. Two other proofs are given in Chapter 5 of this handbook. One is quite close to van Benthem's original approach, the other is based on games. ❑

Nowadays many different proofs are known for this result, and for various extensions and variants. For example, Rosen [109] showed that the result holds over finite models; this is far from obvious, as the restriction to finite models means that many standard results of first-order model theory (such as the Compactness Theorem) cannot be applied. And Otto [99] showed that the modal equivalent guaranteed to exist by the previous theorem can be restricted to a formula of modal operator depth 2^k, where k is the quantifier depth of $\varphi(x)$.

Basic modal logic and first-order logic are analogous in many ways. As we mentioned in Section 2, via the standard translation modal logic immediately inherits basic meta-theoretic properties of its more powerful neighbour, such as the Compactness and Löwenheim-Skolem Theorems. But not all such transfer is automatic. Consider, for example, the *Craig Interpolation* property:

> *If $\varphi \models \psi$ then there exists a formula θ whose vocabulary is included in that of both φ and ψ such that $\varphi \models \theta$ and $\theta \models \psi$.*

Does the same result hold for basic modal formulas φ and ψ such that $\varphi \models \psi$? Appealing to the result for first-order logic gives us a first-order formula θ such that $\mathrm{ST}_x(\varphi) \models \theta$ and $\theta \models \mathrm{ST}_x(\psi)$. But what guarantees that this interpolant is modally definable? Interpolation does in fact hold for the basic modal language (for a detailed account, see Chapter 8 of this handbook),

but additional work is needed to prove this. Nonetheless, interpolation does mesh well with the above preservation results; here is an improvement on the Modal Characterisation Theorem. We say that a first-order formula φ *implies* ψ *along bisimulation* if the following implication holds: if E is a bisimulation between $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$, and $\mathfrak{M}, s \models \varphi$, then $\mathfrak{N}, t \models \psi$.

THEOREM 14 (Modal Characterisation-Interpolation Theorem). *The following are equivalent for all first-order formulas $\varphi(x)$:*

1. *$\varphi(x)$ implies $\psi(x)$ along bisimulation.*

2. *There is a modally definable θ in the common vocabulary of φ and ψ such that $\varphi \models \theta$ and $\theta \models \psi$.*

Proof. The proof can be found in Barwise and van Benthem [11]. Note that the Modal Characterisation Theorem follows by taking $\varphi(x)$ equal to $\psi(x)$. This result does not imply ordinary modal interpolation as it stands: additional work is again needed. ❑

Behind the above observations is the fact that the cheaply transferred properties are universal in some sense, whereas the universal-existential property of interpolation requires honest work. Even so, there is an intuition (based on decades of positive experience with transferring results) that modal logic and first-order logic share all general meta-properties except decidability. No proofs of significant formulations of this idea have been found so far, but we can point to some broad analogies regarding methods. Generally speaking, bisimulation plays the same role for modal logic that potential isomorphism does for first-order logic. This can even be made precise in the following sense. To each first-order model $\mathfrak{M}$ we can associate a modal model whose points are the variable assignments into $\mathfrak{M}$, and whose accessibility relations are changes from one assignment g to another $g(x := d)$ that resets the value for the variable x to the object $d \in \mathfrak{M}$. Then two models $\mathfrak{M}$ and $\mathfrak{N}$ have a potential isomorphism between them iff their associated modal models are bisimilar; see van Benthem [136] for details.

We conclude this discussion with two general results that allow us to switch between modal and first-order relations between models. In essence, both results have the form of a commutative diagram.

LEMMA 15 (First Lifting Lemma). *The following are equivalent for all models $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$:*

1. *$(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$ are modally equivalent.*

2. *$(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$ have elementary extensions to models $(\mathfrak{M}^+, s)$ and $(\mathfrak{N}^+, t)$ which are bisimilar.*

LEMMA 16 (Second Lifting Lemma). *The following are equivalent for all models $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$:*

1. *$(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$ are modally equivalent.*

2. *$(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$ are bisimilar to models $(\mathfrak{M}^+, s)$ and $(\mathfrak{N}^+, t)$ which are elementarily equivalent.*

Proof. The first lifting lemma was originally proved in van Benthem [128]. It is the key item in (some proofs of) the Characterisation Theorem (the $^+$-models are suitably saturated elementary extensions which allow the Characterisation Theorem to be proved rather straightforwardly). The second lifting lemma (see van Benthem [134] for the original result, and Andréka, van Benthem, and Németi [5] for full proof details) involves judicious tree unraveling of the two models, duplicating sub-trees to create uniformity, coupled with an Ehrenfeucht-Fraïssé argument to establish elementary equivalence. ❑

3.6 Bisimulation as a game

Bisimulation can naturally be thought of as a form of process equivalence, but a more dynamic perspective is also possible. We have already seen that the modal satisfaction definition can be recast in the form of a game (recall Figure 6) but the task of determining whether two models are bisimilar can also be viewed in this way. Consider a game between Spoiler (the difference player) and Duplicator (the similarity player) comparing successive pairs in two pointed models $(\mathfrak{M}, w)$ and $(\mathfrak{N}, w')$:

> *If w and w' do not agree on atomic information, Spoiler wins the game in zero rounds. In subsequent rounds, Spoiler chooses a state in one model which is a successor of the current w or w', and Duplicator responds with a matching successor in the other model. If the chosen points differ in their atomic properties, Spoiler wins. If one player cannot move, the other wins. Duplicator wins on infinite runs on which Spoiler does not win.*

This game captures the zigzag behaviour of bisimulations in an obvious sense. It is also *determined*: one of the two players has a winning strategy. (This is because it is an open Gale-Stewart game in the sense of game theory.) For example, returning yet again to the models $\mathfrak{M}$, $\mathfrak{N}$ and $\mathfrak{K}$ considered at the start of this section, we see that Duplicator has a winning strategy in the comparison game for the models $\mathfrak{M}$ and $\mathfrak{K}$ starting from their matched designated points, while Spoiler has one for $\mathfrak{M}$ and $\mathfrak{N}$. The following result clarifies the role of these games precisely:

LEMMA 17 (Adequacy of Modal Comparison Games).

1. *There is an explicit correspondence between Spoiler's winning strategies in a k-round comparison game between $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$ and modal formulas of modal operator depth k on which s and t disagree.*

2. *There is an explicit correspondence between Duplicator's winning strategies over an infinite-round comparison game between $(\mathfrak{M}, s)$ and $(\mathfrak{N}, t)$ and the set of all bisimulations between $\mathfrak{M}$ and $\mathfrak{N}$ that link the points s and t.*

Proof. This result is essentially a fine-grained restatement of the Lemma 9 from a game-theoretic perspective. See Chapter 5 of this handbook for more on game-based approaches to bisimulation. ❑

For example, in the game between the models $\mathfrak{M}$ and $\mathfrak{K}$ given earlier, Duplicator wins by choosing responses that stick to the bisimulation links. And in the game between $\mathfrak{M}$ and $\mathfrak{N}$, Spoiler can win in at most three rounds by using the earlier modal difference formula $\Box(\Box\bot \vee \Diamond\Box\bot)$ of modal operator depth three. In each round he can make sure that some modal difference remains at the current match, with the modal operator depth descending each time.

4 COMPUTATION AND COMPLEXITY

We view modal logic as a tool for representing and reasoning about graphs. Our discussion of expressivity has given us some insight into the representational capabilities of modal logic (at least at the level of models) but what about reasoning?

In this section we discuss modal reasoning from a computational perspective. We concentrate on the *model checking task* and the *satisfiability and validity* problems, but also make some remarks about the *global satisfiability* and the *model comparison* tasks. As we shall see, the complexity of the modal version of these tasks is lower than that of their first-order counterparts.

Before going further, two general remarks. First, although we are about to study reasoning, we are not about to embark on the study of modal proof systems (apart from anything else, the standard proof systems are only relevant to satisfiability and validity checking, and there is more to modal reasoning than this). Secondly, although we are ostensibly moving on from expressivity issues to computational issues, the two topics are intertwined. In essence, the positive computational results reported here arise from negative expressivity results (for example, the inability of the basic modal language to force the existence of infinite models).

4.1 Model checking

The model checking task can be formulated locally:

> *Given a (finite) model $\mathfrak{M}$, a point w in $\mathfrak{M}$, and a basic modal formula φ, is φ satisfied in $\mathfrak{M}$ at w?*

Or globally:

> *Given a (finite) model $\mathfrak{M}$, and a basic modal formula φ, is φ satisfied at all points in $\mathfrak{M}$?*

Or in a form that subsumes both the local and global perspectives:

> *Given a (finite) model $\mathfrak{M}$, and a basic modal formula φ, return the set of points in $\mathfrak{M}$ that satisfy φ.*

In what follows we shall work with the last formulation, which is probably the most common way of thinking about model checking in practice.

Now, model checking is clearly a task with computational content — but is it really a *reasoning* task? In our view, yes. In essence, a model is a 'flat' store of information: it consists of a collection of entities, together with a specification of which entities have which properties, and which entities are related by which atomic relations. A modal formula, on the other hand, is a recursively constructed tree. The embedding of connectives and modalities within one another permits relatively short formulas to make interesting assertions, assertions that go way beyond the mere listing of atomic facts. If we add to these differences the practical observation that in typical applications the formula will be much smaller than the model, we see that model checking is about synchronising two very different forms of information: it tells us whether the abstract information embodied in the formula is implicitly present in the model, and gives us a set of points where this implicit information emerges. Viewed this way, model checking is a quintessential reasoning task.

Moreover, model checking has turned out to be of great practical importance — indeed, one of the more salutary lessons computer science has taught logic is just how important this modest looking form of reasoning actually is. Nowadays the practical importance of modal model

checking dwarfs that of determining modal satisfiability or validity (the tasks logicians have traditionally viewed as paramount) as a wide range of practical tasks can be modeled in a computationally natural manner, and efficiently solved, via model checking. A classic example is hardware verification. Even though a computer chip is a concrete object, it gives rise to a natural abstract model, namely the set of all states the chip can be in, and the transitions between them. If a chip is to work satisfactorily, its computational runs (that is, the sequences of states it can follow by making transitions from the initial state) should possess a number of high-level 'emergent' properties: for example, these runs should not enter deadlock situations. If we have a modal language that can express the desired properties (for example, absence of deadlock) then by checking the formula in the model representing the chip we can determine whether the design is satisfactory or not.

So how should we perform model checking? The standard approach is to use a bottom-up *labeling algorithm*. To model check a formula φ we label every point in the model with all the subformulas of φ that are true at that point. We start with the proposition symbols: the valuation tells us where these are true, so we label all the appropriate points. We then label with more complex formulas. The booleans are handled in the obvious way: for example, we label w with $\psi \wedge \theta$ if w is labeled with both ψ and θ. As for the modalities, we label w with $\Diamond\varphi$ if one of its R-successors is labeled with φ, and we label it with $\Box\varphi$ if all of its R-successors are labeled with φ. A precise definition of the algorithm for checking diamond formulas is given in the pseudo-code of Figure 15.

procedure $\mathit{Check}\Diamond(\psi)$
    $T := \{v \mid \psi \in \mathit{label}(v)\}$;
    while $T \neq \emptyset$ **do**
        choose $v \in T$;
        $T := T \setminus \{v\}$;
        for all w **such that** Rwv **do**
            if $\Diamond\psi \notin \mathit{label}(w)$ then
            $\mathit{label}(w) := \mathit{label}(w) \cup \{\Diamond\psi\}$;
            end if ;
        end for all ;
    end while ;
end procedure

Figure 15. Model checking $\Diamond\psi$.

The beauty of this algorithm is that we never need to duplicate work: once a point is labeled as making φ true, that's it. This makes the algorithm run in time polynomial in the size of the input formula and model: the algorithm takes time of the order of

$$\mathit{con}(\varphi) \times \mathit{nodes}(\mathfrak{M}) \times \mathit{nodes}(\mathfrak{M}),$$

where $\mathit{con}(\varphi)$ is the number of connectives in φ, and $\mathit{nodes}(\mathfrak{M})$ is the number of nodes in $\mathfrak{M}$. To see this, note that $\mathit{con}(\varphi)$ tells us how many rounds of labeling we need to perform, one of the $\mathit{nodes}(\mathfrak{M})$ factors is simply the upper bound on the nodes that need to be labeled, while the other is the upper bound on the number of successor nodes that need to be checked.

Thus modal model checking is a computationally tractable task, but this is not the case for first-order logic. In fact, model checking first-order formulas is a PSPACE-complete task (see

Chandra and Merlin [22]). That is, although it is possible to write an algorithm that solves the first-order model checking task using an amount of computer memory that is only polynomial in the size of the input model and formula, the algorithm may require running time that is exponential in the size of the input. The problem, of course, lies with the quantifiers. Assuming that the standard assumptions made in complexity theory are correct, there is no way of adapting the labeling algorithm (or indeed, any other algorithm) to perform first-order model checking in polynomial time.

However the labeling algorithm sketched above *does* adapt to more powerful modal languages, and this is important. As we said above, when model checking we want to state interesting high-level properties of the situation we are modeling, and often the ordinary $\Box$ and $\Diamond$ modalities simply aren't expressive enough. In model checking applications, it is usual to work with tree-like models, namely trees of computational runs. On such models $\Diamond$ is interpreted as "at some immediate successor state". This is natural, to be sure, but somewhat limited. However, by adding the binary Until modality, we gain access to entire *sequences* of successor states:

$$\begin{array}{lll} \mathfrak{M}, s \models U(\psi,\theta) & \text{iff} & \text{there is a } t \text{ such that } sR^*t \text{ and } \mathfrak{M}, t \models \psi, \\ & & \text{and for all } u \text{ such that } sR^*u \text{ and } uR^+t \text{ we have } \mathfrak{M}, u \models \theta. \end{array}$$

Here R^* is the reflexive transitive closure of the "immediate successor" transition relation R explored by $\Diamond$, and R^+ is its transitive closure. Thus Until gives us a direct handle on the computational runs that can be followed in the model, and this clearly places interesting expressive power at our disposal. Nowadays the Until modality is a fundamental component of some of the most important model checking formalisms — formalisms such as LTL (Linear Time Temporal Logic) and CTL (Computational Tree Logic). For an introduction to these logics, see Chapter 11 of this handbook, or Clarke, Grumberg and Peled [25].

We shall examine the Until operator and the extra expressivity it offers more closely in Section 6.3. Here we simply want to address the following question: how do we extend the labeling algorithm to handle formulas of the form $U(\psi,\theta)$? Here's the basic idea. First, if any point w is labeled with ψ, label w with $U(\psi,\theta)$. Second, if any point v is labeled with θ and at least one R-successor of v is labeled with $U(\psi,\theta)$, then label v with $U(\psi,\theta)$. It should be clear that these two steps reflect the semantics for Until just given; the pseudo-code given in Figure 16 shows how to make the basic idea precise.

Now for an important point. Throughout the previous discussion we have tacitly assumed that we have some way of representing formulas and finite models that is suitable for computational implementation. It is probably not worth sketching details of how this might be done: nowadays it seems safe to assume that most readers of a technical book on logic have at least a nodding acquaintance with programming (indeed, we suspect that most of our readers would find it straightforward to devise a computational syntax for models and modal languages, and to implement simple programs for working with them).

Nonetheless, such issues cannot be taken lightly. A major factor in the spectacular progress of model checking has been the development of *Binary Decision Diagrams* (BDDs) and *Ordered Binary Decision Diagrams* (OBDDs). BDDs (which are compact representations of boolean expressions) were introduced by Lee [88] and Akers [3], and OBDDs (a more sophisticated form of BDD with fewer representational redundancies) were introduced by Bryant [17]. BDDs were first proposed for model checking by Burch, Clarke, McMillan, Dill, and Hwang [18] and as the title of this paper indicates ("Symbolic model checking: 10^{20} states and beyond") this led to a dramatic increase in the size of the models that could be handled. It is important not to underestimate the gap between the labeling algorithm sketched above, and what it takes to make

```
procedure CheckU(ψ, θ)
    T := {v | ψ ∈ label(v)} ;
    for all v ∈ T do
        label(v) = label(v) ∪ {U(ψ, θ)} ;
    end for all ;
    while T ≠ ∅ do
        choose v ∈ T ;
        T := T \ {v} ;
        for all w such that Rwv do
            if U(ψ, θ) ∉ label(w) and θ ∈ label(w) then
                label(w) := label(w) ∪ {U(ψ, θ)} ;
                T := T ∪ {w} ;
            end if ;
        end for all ;
    end while ;
end procedure
```

Figure 16. Model checking $U(\varphi, \theta)$.

a working model checker handle a large model. Crossing this gap requires a combination of theoretical insight and computational expertise, and an entire research community is devoted to exploring the issues involved.

For a good textbook level introduction to model checking, see Huth and Ryan [72]. This book not only introduces the basic algorithms, it also shows how they can be implemented with the aid of OBDDs. Moreover, it discusses modal checking for the modal μ-calculus (which we introduce in Section 6.7). For a more advanced treatment, see Clarke, Grumberg and Peled [25]. Finally, for an account of model checking via automata-theoretic methods, see Chapter 17 of this handbook.

4.2 *Satisfiability and validity: decidability*

It is often said that modal logic is decidable. This can be read as shorthand for the following claim: the *validity problem* for the basic modal language (*given a basic modal formula φ, is φ valid?*) is decidable. That is, it is possible (ignoring constraints of time and space) to write a computer program which takes a basic modal formula as input, and halts after a finite number of steps and correctly tells us whether it is valid or not.

The decidability of modal logic can also be viewed as a claim that the *satisfiability problem* for the basic modal language (*given a basic modal formula φ, is φ satisfiable in some model?*) is decidable. That is, it is possible (again, ignoring constraints of time and space) to write a computer program which takes a basic modal formula as input, and halts after a finite number of steps and correctly tells us whether it is satisfiable in some model or not. The validity and satisfiability problems are *dual problems*: a modal formula φ is valid iff $\neg\varphi$ is not satisfiable, hence if we have a method for solving one problem, we have a method for solving the other. In what follows we show that both problems are decidable; we'll talk in terms of satisfiability.

A lot is known about the decidability of satisfiability problems for various logics, so it is not too difficult to establish modal decidability: we can do so by reducing the problem to known

results for other logics. Here's an easy example. The satisfiability problem for the *two variable fragment* of first-order logic (that is, the fragment of first-order logic in which every formula contains only two variables) is decidable. Now, every basic modal formula can be translated into a formula in the two-variable fragment. To see this we need simply make a small adjustment to the standard translation ST_x. Whenever we translate a $\Diamond$ or a $\Box$, instead of choosing a completely new variable to quantify over accessible points, we use a second fixed variable (say y). If we later encounter another $\Diamond$ or $\Box$, we flip back to the original variable x, and so on. More precisely, we redefine ST_x so it always uses y to quantify over accessible points, and define a twin translation ST_y which always quantifies using x. Here are the key clauses:

$$\begin{array}{ll} \mathrm{ST}_x(\Diamond\varphi) = \exists y\,(Rxy \wedge \mathrm{ST}_y(\varphi)) & \mathrm{ST}_y(\Diamond\varphi) = \exists x\,(Ryx \wedge \mathrm{ST}_x(\varphi)) \\ \mathrm{ST}_x(\Box\varphi) = \forall y\,(Rxy \rightarrow \mathrm{ST}_y(\varphi)) & \mathrm{ST}_y(\Box\varphi) = \forall x\,(Ryx \rightarrow \mathrm{ST}_x(\varphi)). \end{array}$$

The interleaving of ST_x and ST_y guarantees that for any basic modal formula φ, $\mathrm{ST}_x(\varphi)$ will contain only the two variables x and y, and it should be clear that the modified translation is equivalent to the original one. It follows that the satisfiability problem for the basic modal language must be decidable: to test a modal formula for satisfiability, simply translate it with this new version of the standard translation, and then apply the satisfiability algorithm for the two-variable fragment to the output.

It is pleasant that modal decidability can be established so easily, but the proof isn't particularly instructive. The following semantic argument is somewhat more revealing. We shall show that the basic modal language has the *finite model property*, or to put it another way, that it does not have the expressive strength required to force the existence of infinite models. Needless to say, this is in sharp contrast with first-order logic: even such a simple first-order formula as

$$\forall x \neg Rxx \wedge \forall xyz(Rxy \wedge Ryz \rightarrow Rxz) \wedge \forall x \exists y Rxy$$

has only infinite models. In fact, the basic modal language has a rather strong form of the finite model property. We shall show the following:

THEOREM 18 (Strong Finite Model Property). *Let φ be a basic modal formula. If φ is satisfiable, then it is satisfiable on a finite model containing at most $2^{s(\varphi)}$ points, where $s(\varphi)$ is the number of subformulas of φ.*

The decidability of the modal satisfiability problem follows immediately from this result. If a modal formula φ is satisfiable at all, it is satisfiable on a model containing at most $2^{s(\varphi)}$ points. As there are (up to isomorphism) only finitely many such models, exhaustive (and exhausting!) search through them all will settle the issue of φ's satisfiability.

Just as important as the result is the method we shall use to prove it: *filtrations*. These are a standard item in the modal logician's toolkit, and have been used to prove completeness and decidability results for many different modal systems. The basic idea underlying the method is simplicity itself: given a modal formula φ and a model $\mathfrak{M}$ that satisfies it, we make a finite model $\mathfrak{M}$ by collapsing to a single point all the points within $\mathfrak{M}$ that satisfy the same subformulas of φ. But there is a tricky issue: how should we define the relation on the collapsed points in such a way that φ remains true in the finite model? Let's work through the details and see.

We shall say that a set of basic modal formulas Σ is *subformula closed* if every subformula of every formula in Σ is a member of Σ (that is, if $\varphi \wedge \psi \in \Sigma$ then so are φ and ψ, and if $\neg\varphi \in \Sigma$ then so is φ; and if $\Box\varphi \in \Sigma$, then so is φ, and so on). We now define:

DEFINITION 19 (Filtrations). Let $\mathfrak{M} = (W, R, V)$ be a model, let Σ be a subformula closed set of formulas, and let $\leftrightsquigarrow_\Sigma$ be the equivalence relation on the states of $\mathfrak{M}$ defined as follows:

$$w \leftrightsquigarrow_\Sigma v \text{ iff for all } \varphi \text{ in } \Sigma\text{: } (\mathfrak{M}, w \models \varphi \text{ iff } \mathfrak{M}, v \models \varphi).$$

The official notation for the equivalence class of a point w of $\mathfrak{M}$ with respect to $\leftrightsquigarrow_\Sigma$ is $|w|_\Sigma$, but in what follows we'll usually assume that Σ is clear from context and simply write $|w|$.

Let $W_\Sigma = \{|w| \mid w \in W\}$. Suppose $\mathfrak{M}^f_\Sigma$ is any model (W^f, R^f, V^f) such that:

1. $W^f = W_\Sigma$.
2. If Rwv then $R^f|w||v|$.
3. If $R^f|w||v|$ then for all $\Diamond\varphi \in \Sigma$, if $\mathfrak{M}, v \models \varphi$ then $\mathfrak{M}, w \models \Diamond\varphi$.
4. $V^f(p) = \{|w| \mid \mathfrak{M}, w \models p\}$, for all proposition symbols p in Σ.

Then $\mathfrak{M}^f_\Sigma$ is called a filtration of $\mathfrak{M}$ through Σ. In what follows we'll drop the subscripts and write $\mathfrak{M}^f$ instead of $\mathfrak{M}^f_\Sigma$.

Two points should be made about this definition. First, observe $\mathfrak{M}^f$ is a filtration of $\mathfrak{M}$ through a subformula closed set of formulas Σ, then $\mathfrak{M}^f$ contains at most $2^{|\Sigma|}$ nodes, where $|\Sigma|$ is the cardinality of Σ. This should be clear: after all, the points of $\mathfrak{M}^f$ simply are the equivalence classes in W_Σ, and there cannot be more than $2^{|\Sigma|}$ of these. Second, the previous definition does *not* specify an accessibility relation on W_Σ — it only imposes constraints (namely clauses 2 and 3) on the properties a suitable accessibility relation R^f should have. That the constraints imposed are sensible is shown by the following result:

THEOREM 20 (Filtration Theorem). *Let $\mathfrak{M}^f$ $(= (W_\Sigma, R^f, V^f))$ be a filtration of $\mathfrak{M}$ through a subformula closed set of basic modal formulas Σ. Then for all formulas $\sigma \in \Sigma$, and all nodes w in $\mathfrak{M}$, we have $\mathfrak{M}, w \models \sigma$ iff $\mathfrak{M}^f, |w| \models \sigma$.*

Proof. By induction on the structure of formulas. The case for proposition symbols is immediate from the definition of V^f, and because Σ is closed under subformulas, the inductive step for the boolean connectives is clear.

So suppose $\Diamond\sigma \in \Sigma$ and $\mathfrak{M}, w \models \Diamond\sigma$. Then there is a v such that Rwv and $\mathfrak{M}, v \models \sigma$. As $\mathfrak{M}^f$ is a filtration, by the first constraint on R^f (clause 2 of the previous definition) we have that $R^f|w||v|$. As Σ is subformula closed, $\sigma \in \Sigma$, hence by the inductive hypothesis $\mathfrak{M}^f, |v| \models \sigma$. Hence $\mathfrak{M}^f, |w| \models \Diamond\sigma$.

Conversely, suppose $\Diamond\sigma \in \Sigma$ and $\mathfrak{M}^f, |w| \models \Diamond\sigma$. Then there is a state $|v|$ in $\mathfrak{M}^f$ such that $R^f|w||v|$ and $\mathfrak{M}^f, |v| \models \sigma$. As $\sigma \in \Sigma$, by the inductive hypothesis $\mathfrak{M}, v \models \sigma$. Making use of the second constraint on R^f (clause 3 of the previous definition) yields $\mathfrak{M}, w \models \Diamond\sigma$. ❑

It only remains to verify that relations satisfying the constraints demanded of R^f actually exist. They do. Define:

1. $R^s|w||v|$ iff $\exists w' \in |w|\ \exists v' \in |v|\ Rw'v'$.
2. $R^l|w||v|$ iff for all formulas $\Diamond\varphi$ in Σ: $\mathfrak{M}, v \models \varphi$ implies $\mathfrak{M}, w \models \Diamond\varphi$.

It is straightforward to show that both relations satisfy the required constraints. Actually, you can show a little more: if R^f is any relation satisfying the above constraints then $R^s \subseteq R^f \subseteq R^l$. For this reason, R^s and R^l are said to give rise to the smallest and largest filtrations respectively.

So we have proved Theorem 18: the basic modal language indeed has the strong finite model property. As we argued above, this in turn establishes the decidability of the basic modal satisfiability problem. Now, as is well known, the satisfiability problem for full first-order logic is undecidable. First-order logic is the classic example of a language where expressivity has been purchased at the expense of decidability. The basic modal language reverses this trade-off.

4.3 Satisfiability and validity: complexity

What do the decidability proofs just given tell us about the computational complexity of the modal satisfiability problem? Only that it can be solved in NEXPTIME (that is, non-deterministic exponential time). This is clear from the filtration proof: to see if φ is decidable, we can non-deterministically choose a model containing at most $2^{s(\varphi)}$ points, and then check whether or not it satisfies φ. As we have seen from our discussion of model checking, the checking takes time polynomial in the size of model; however as the model is exponential in the size of the input formula φ, this is a complex task. The reduction to the satisfiability problem for the two-variable fragment yields the same upper bound, as this problem is NEXPTIME-complete.

But the satisfiability problem for basic modal logic is PSPACE-complete. That is, given a modal formula φ, it is possible to write an algorithm to determine whether or not φ is satisfiable that uses an amount of computer memory that is only polynomial in the size of φ. Now, most complexity theorists believe that PSPACE-complete problems are harder than the satisfiability problem for classical propositional logic (the classic NP-complete problem) but easier than EXPTIME-complete problems, which in turn are believed to be easier than NEXPTIME-complete problems. So, given standard complexity-theoretic assumptions, the modal satisfiability problem is probably easier than our earlier decidability proofs suggest.

How do we design a PSPACE algorithm for modal satisfiability? We cannot give a detailed answer here, but we can point to an expressive weakness of modal logic which should make it plausible that PSPACE algorithms for modal satisfiability exist.

LEMMA 21. *Let $\mathfrak{M} = (W, R, V)$ be a model, let $w \in W$, let n be a natural number, let $S_{n,w}$ be the subset of W containing w and all points in W reachable from w by making at most n R-transitions, and let $\mathfrak{N}$ be the submodel $(S_{n,w}, R|_S, V|_S)$, where $R|_S$ and $V|_S$ are the restrictions of R and V respectively to $S_{n,w}$. Then, for all basic modal formulas φ such that md$(\varphi) \leq n$, we have that $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{N}, w \models \varphi$.*

That is, if we take a model $\mathfrak{M}$, and extract a submodel $\mathfrak{N}$ from it by throwing away all points that are more than n steps away from w, then no formula with modal operator depth of at most n can distinguish the two models at w. Modal formulas have shallow vision. And if we combine this lemma with what we have already learned about finite models and bisimulations, we obtain the following:

THEOREM 22. *Every formula φ in the basic modal language is satisfiable in a model based on a finite tree of depth at most md(φ).*

Proof. As model logic has the finite model property, if a modal formula is satisfiable, it is satisfiable on a finite model $\mathfrak{M}$ at some point w. As we remarked in the previous section, it is always possible to unravel a model into an equivalent tree-based model. Now, if we unravel $\mathfrak{M}$ about w, we don't necessarily obtain a finite model, but (as $\mathfrak{M}$ is finite) we do obtain a model

based on a tree with a finite branch factor, and this model satisfies φ at its root. If we then chop off all points more than $md(\varphi)$ away from the root we obtain a finite model which (by the previous lemma) satisfies φ at its root. ❑

So every modal formula is satisfiable on a shallow tree, and we are now in a position to appreciate how PSPACE algorithms for modal satisfiability work. In essence, they construct shallow trees branch by branch. If a branch is successfully constructed (something which takes only space polynomial in the size of the input formula, as the length of the branch is bounded by $md(\varphi)$) the branch is discarded (thus freeing up the memory) and the next branch is then constructed. There may be many branches, so it may take exponential time to construct them all, but as all branches are discarded once they are constructed, such an algorithm uses only polynomial space. This sketch has neglected some important issues (such algorithms require space for recording book-keeping details, and we need to ensure that the space used for this is not excessive) but it does describe, in broad terms, how many modal satisfiability algorithms (notably those based on tableaux or games) work.

But we should issue a word of warning: it's not always so easy. Yes, matters are relatively straightforward here, but that is because we have been working with the *basic* modal language over the class of *all* models. If we impose restrictions on the class of models we are working with (as we shall do in Section 5) or work with richer modal languages (as we shall do in Section 6), or both, we can easily find ourselves faced with undecidable, or even highly undecidable, satisfiability and validity problems.

4.4 Other reasoning tasks

We have discussed the big three (model checking, and satisfiability and validity checking) but this by no means exhausts the reasoning tasks of interest. To conclude this section, let's briefly consider two others.

Although we have stressed the locality of modal logic, some problems demand a global perspective. In particular, if we view a modal formula as a general background *constraint*, we will typically want it to be globally satisfied: that is, we will be interested in models $\mathfrak{M}$ such that $\mathfrak{M} \models \varphi$. The importance of the global satisfiability problem has been strongly emphasised by the description logic community. Indeed, description logic builds into its architecture the idea of a *Terminological Box* (or *TBox*), a collection of formulas that encode background knowledge about some domain (for example, that all men are mortal, that all Martians own flying saucers, or that each employee has a social security number). Description logicians are interested in models in which the TBox is globally satisfied, for these are the models that reflect all the background assumptions.

Once the importance of background constraints is realised, it becomes clear that it is not the pure global satisfiability task itself that is of primary interest. Rather, it is the *local-global satisfiability task*: given formulas φ and ψ, is there a model which locally satisfies φ and globally satisfies ψ? That is, is it possible to satisfy φ subject to the global constraint ψ?

Here's an example. Suppose we're working in a zoological setting, and are interested in the interaction of maternal love and professional responsibility on the feeding of our furry ursine brethren. To put it another way, suppose we have the following TBox:

$$\begin{array}{rl} \mathit{bear} \vee \mathit{human} & \qquad \mathit{bear} \rightarrow \langle \text{MOTHER} \rangle \mathit{bear} \\ \mathit{bear} \rightarrow \neg \mathit{human} & \qquad \mathit{bear} \rightarrow [\text{FEDBY}](\mathit{zoo\text{-}keeper} \vee \mathit{mother}) \end{array}$$

Let's call this TBox BEAR-CARE. The sort of queries we might be interested in posing are: is it possible to globally satisfy BEAR-CARE and, simultaneously, to locally satisfy

$$\langle \text{MOTHER} \rangle (\textit{bear} \wedge \textit{human})?$$

(No, it's not.) And is it possible to globally satisfy BEAR-CARE and simultaneously to locally satisfy

$$\langle \text{FEDBY} \rangle (\neg \textit{human} \wedge \neg \textit{mother})?$$

(Yes, it is: BEAR-CARE doesn't rule out having bears as zoo-keepers. This may well be a bug in the TBox.)

Local-global satisfiability problems are also natural in the setting of parsing problems. It is possible to encode various kinds of grammars (such as regular grammars or context-free grammars) as modal formulas (see Chapter 19 of this handbook for a discussion of such approaches). Then, given a string of symbols, the parsing problem is to decide whether it is possible to find a model which embodies all the constraints encoded in the grammar, and which simultaneously satisfies the formula encoding the input string. That is, we would like to globally satisfy the modal formula GRAMMAR and simultaneously locally satisfy INPUT-STRING.

Unsurprisingly, both the global, and the local-global satisfiability tasks are tougher than the ordinary satisfiability problem:

THEOREM 23. *The global satisfiability and the local-global satisfiability tasks for basic modal languages are both EXPTIME-complete.*

Proof. The stated result is an immediate consequence of Hemaspaandra's [118, 65] complexity results for the universal modality (we introduce the universal modality in Section 6.1). But the result holds for even stronger languages; see De Giacomo and Lenzerini [28] for related results for more expressive description logics. ❑

EXPTIME-complete problems are decidable but provably intractable: they contain problem instances that will require time exponential in the size of the input to solve (which can mean that they require more time than the expected lifetime of the universe). This, however, is a worst-case scenario. One of the most important recent developments in computational logic has come from the description logic community, who have shown it is possible to specify and implement tableaux-based algorithms for such problems that are remarkably efficient in practice. Moreover, interesting work exists on performing modal theorem proving via (non-standard) translations into first-order logic, so that optimised first-order resolution provers can be applied to the task. For a detailed discussion and comparison of these methods, see Chapter 4 of this handbook, and for a deeper examination of the complexity of modal logic, see Chapter 3.

We conclude with a remark on the *model comparison* task. As bisimulation is the modally fundamental notion of graph equivalence, it is natural to wonder how difficult it is to determine when two models are bisimilar. The corresponding problems for first-order logic (namely, testing for graph isomorphism) is thought to be difficult: there is no known polynomial algorithm for testing for graph isomorphism, though the problem has not been shown to be NP-complete either. In fact, the problem of identifying isomorphic graphs is sometimes regarded as giving rise to a special complexity class of its own.

Testing for bisimulation, however, turns out to be computationally tractable, and there are elegant polynomial algorithms which work by discarding pairs of point that cannot make it into any bisimulation (see Dovier, Piazza and Policriti [33]). Again an expressivity result lies behind this result: the maximal bisimulation between two models $\mathfrak{M}$ and $\mathfrak{N}$ is explicitly definable

in a first-order fixed-point language over the disjoint union $\mathfrak{M} \uplus \mathfrak{N}$ of the two models. Such languages have been studied extensively in computer science, and they are known to have good computational behaviour.

Let us summarise our discussion. For a number of tasks, the basic modal language (interpreted over the class of all models) is computationally better behaved than the corresponding first-order language (interpreted over the same models). Figure 17 summarises the relevant facts (PTIME is short for Polynomial Time). Of course, this better computational behaviour comes about because

	Model Checking	Satisfiability	Model Comparison
FOL	PSPACE-complete	Undecidable	in NP
ML	PTIME	PSPACE-complete	PTIME

Figure 17. First-order logic and modal logic: computational properties summarised.

the basic modal language is not nearly as expressive as first-order logic. Thus the pressing questions are: what are the trade-offs? And can this better computational behaviour be lifted to more expressive modal logics, and (if so) how? We shall revisit these questions in the following two sections.

5 RICHER LOGICS

Until now, we have deliberately said rather little about modal *logics* and what they are. Instead we have acted as if there was only one modal logic of any interest, namely the set of valid formulas (that is, the set of formulas satisfied at all points in all models) or, to put it syntactically, the set of formulas generated by the minimal proof system **K** (which we defined at the start of Section 2.2). But traditional presentations of modal logic tend to emphasise the *multiplicity* of modal logics, and devote a great deal of attention to logics richer than **K**, logics with such names as **T**, **K4**, **S4**, **S5**, **GL**, and **Grz**. Where do richer modal logics come from?

As a first approximation (we'll shortly see why it's only an approximation) we might say that richer logics emerge at the level of *frames*, via the concept of *frame validity*. Let $\varphi(p_1, \ldots, p_n)$ be a basic modal formula built out of the proposition symbols $p_1, \ldots, p_n$. We say that $\varphi(p_1, \ldots, p_n)$ is *valid on a frame* $\mathfrak{F} = (W, R)$ *at a point* w if, for each valuation V for its proposition symbols $p_1,\ldots,p_n$, we have that φ is satisfied in the resulting model at w; in such a case we write $\mathfrak{F}, w \models \varphi$. We say φ is *valid on* $\mathfrak{F}$ if it is valid at each point in $\mathfrak{F}$, and we write this as $\mathfrak{F} \models \varphi$. Moreover, we say that a modal formula is *valid on a class of frames* F if it is valid on each frame $\mathfrak{F}$ in F. Note that a valid formula (as defined in Section 2.1) is simply a formula that is valid on the class of all frames.

The starting point for this section is the observation that different applications of modal logic typically validate different modal axioms, axioms over and above those to be found in the minimal system **K**. For example, if we view our models as flows of time, it is natural to assume that the accessibility relation is transitive, and (as the reader should check) any instance of the schema $\Box\varphi \to \Box\Box\varphi$ is valid on the class of transitive frames (for example, the formula $\Box p \to \Box\Box p$ is valid on such frames, and $\Box(p \vee q) \to \Box\Box(p \vee q)$ is too). However no instance of this schema (which for historical reasons is called 4) is provable in **K**, so if we want a logic for working with temporal flows we should add all its instances as extra axioms, and doing so yields the logic known as **K4**. Or suppose we are modeling situations where the frame relation has to be treated

as a partial function. As the reader should check, all instances of the schema $\Diamond\varphi \rightarrow \Box\varphi$ are valid on the class of such frames, and none of them can be proved in **K**, so once again we should add them as extra axioms. Doing so yields the logic called $\mathbf{KAlt}_1$.

We begin this section by briefly discussing such axiomatic extensions of **K** a little further. But our real interest is not the richer logics that arise by adding extra axioms (for an introduction to this topic, see Chapter 2 of this handbook) rather it centres on the following semantic questions: what can modal formulas say about frames, and how do they say it? As we shall see, there is a fundamental expressivity distinction between the level of models and the level of frames: whereas modal logic at the level of models is the bisimulation invariant fragment of first-order logic, at the level of frames it is a fragment of second-order logic.

5.1 Axioms and relational frame properties

One of the most attractive features of modal logic is the illumination provided by the fact that modal axioms reflect properties of accessibility relations. A typical modal completeness theorem reads like this:

THEOREM 24. *A formula is provable in* ***S4*** *iff it is true in all models based on frames whose accessibility relation is transitive and reflexive.*

Proof. See Chapter 2 of this handbook (or indeed, virtually any introduction to modal logic). ❑

That is, the theorems of **S4** are true in all graphs with a transitive and reflexive relation, while its non-theorems have some transitive and reflexive counter-model; the additional axioms reflect simple visualisable geometric conditions in the semantics. There are many techniques for proving such completeness results, ranging from simple inspection of the *canonical model* constructed from all complete theories in the logic (this fundamental technique is introduced in Chapter 2 of this handbook) to various types of model surgery (such as filtration, unraveling, and taking bounded morphic images). Moreover, the motivations for proving modal completeness theorems may differ. Sometimes we start with an independently interesting proof system and try to find a useful corresponding class of frames. The classic example of this is the proof system **GL**, that is **K** enriched with all instances of the Löb axiom schema $\Box(\Box\varphi \rightarrow \varphi) \rightarrow \Box\varphi$, which arose via the study of arithmetical provability (see Chapters 2 and 16 of this handbook for further discussion of **GL**) and was later proved complete with respect to the class of finite trees (where the binary relation interpreting the modalities is the transitive closure of the one-step daughter-of tree relation). Sometimes, however, we might start with a natural model class — say an interesting space-time structure — and try to axiomatise its modal validities. The literature is replete with both variants.

Nowadays a lot is known about axiomatic extensions of **K**. For a start, it turns out that there are uncountably many such *normal modal logics*, as they are often called. It is usual to identify a normal modal logic with the set of formulas it generates, and we say that a modal logic is consistent if it does not contain all formulas. This identification immediately induces a lattice structure on the set of all such logics. The cartography of this landscape is an object of study in its own right; here we shall only mention that, because of the following result, it contains two major highways.

THEOREM 25. *Let* ***Id*** *be the normal modal logic generated by* ***K*** *enriched with all instances of the axiom schema $\varphi \leftrightarrow \Box\varphi$, and let* ***Un*** *be the normal modal logic generated by* ***K*** *enriched with the axiom $\Box\bot$. Every consistent normal modal logic is either a subset* ***Id*** *or* ***Un****.*

Proof. See Makinson [92] for the original (algebraic) proof. After we have introduced generated submodels and bounded morphisms for frames we will be able to sketch the semantic ideas that underly this result, and we shall do this shortly. ❑

Now, as the reader should check, every instance of $\varphi \leftrightarrow \Box\varphi$ is valid on frames which consist of a collection of isolated reflexive points, and $\Box\bot$ is valid on frames consisting of a collection of isolated irreflexive points. Moreover, using standard techniques it is easy to show that **Un** is complete with respect to the first frame class, and **Id** with respect to the second. Thus the semantic content of Theorem 25 is that every normal modal logic is contained in the logic of one of these frame classes; for example, **S4** lies on the first road, and **GL** on the second.

But the most important fact to have emerged about normal modal logics is that *not* all of them have frame-based characterisations. In fact, frame completeness results (such as the result for **S4** noted above) are the exception rather than the rule. Thus our earlier remark that richer logics emerged at the level of frames via the concept of frame validity was very much a first approximation: the notion of frame validity simply does not provide an adequate semantic basis for studying all normal modal logics. Here is a concrete example of a *frame incompleteness* result:

THEOREM 26. *Let* ***TMEQ*** *be the normal modal logic obtained by enriching* ***K*** *with all instances of the following schemas:* $\varphi \to \Diamond\varphi$ *(**T**),* $\Box\Diamond\varphi \to \Diamond\Box\varphi$ *(**M**),* $\Diamond(\Diamond\varphi \wedge \Box\psi) \to \Box(\Diamond\varphi \vee \Box\varphi)$ *(**E**), and* $(\Diamond\varphi \wedge \Box(\varphi \to \Box\varphi)) \to \varphi$ *(**Q**). There is no class of frames that validates precisely the formulas in* ***TMEQ****.*

Proof. See van Benthem [129]. ❑

Such incompleteness results (which were first proved in the early 1970s by Thomason [125] and Fine [43]) were important in the development of modal logic. For a start, they forced modal logicians to examine alternative ways of semantically characterising normal modal logics, and this led to a renaissance in algebraic semantics of modal logic (see Chapter 6 of this handbook for more on this topic). But they also had another effect, one more relevant to the present chapter: they stimulated a wave of semantic research at the level of frames. This new wave of research was centred around the notion of frame definability, the topic to which we now turn.

5.2 Frame definability and undefinability

Before getting to work, a brief remark. There is another way of thinking about axiomatic extensions of **K**. Instead of viewing them as giving rise to brand new modal logics, we can simply view them as *theories* constructed over the minimal logic **K** in much the same way as a first-order theory (of say, linear orders) is constructed over the set of first-order validities. Nothing of substance hangs on this shift of perspective, but it fits more naturally with our focus on expressivity.

So, bearing this in mind, let's pose the first question: what can modal formulas say about frames? A natural way to approach this is to introduce the concept of *frame definability*. We shall say that a modal formula φ defines a class of frames F iff it is valid on precisely the frames in F. That is, not only must φ be valid on every frame in F, it must also be possible to falsify φ on any frame that is not in F. So, what classes of frames can modal languages define? Here are some simple examples:

PROPOSITION 27.

1. $\Box p \to \Box\Box p$ *defines the class of transitive frames; that is, frames such that* $\forall xyz(Rxy \wedge Ryz \to Rxz)$.

2. *$\Diamond p \to \Box p$ defines the class of frames where the frame relation R is a partial function; that is, frames such that $\forall xyz(Rxy \wedge Rxz \to y = z)$.*

3. *$p \leftrightarrow \Box p$ defines the class of frames which consist of isolated reflexive points; that is, frames such that $\forall xy(Rxy \leftrightarrow x = y)$.*

4. *$\Box \bot$ defines the class of frames which consist of isolated irreflexive points; that is, frames such that $\forall xy \neg Rxy$.*

Proof. We have already asked the reader to check that these formulas are valid on the class of frames in question. So to complete the proofs of these definability claims we need merely check that each formula can be falsified on any frame that does not belong to the relevant class.

Let's deal with the second example. Suppose (W, R) is a frame where R is not a partial function. This means that there is a point $w \in W$ that has two distinct R-successors, say u and v. It follows that we can falsify $\Diamond p \to \Box p$ on (W, R) at w. For let V be the valuation that makes p true at u and nowhere else. Then $(W, R, V), w \models \Diamond p$ but $(W, R, V), w \not\models \Box p$, since p is not true at v. So we have falsified $\Diamond p \to \Box p$ on (W, R) as required. ❑

A remark on terminology. Instead of saying, for example, that $\Box p \to \Box\Box p$ defines the class of transitive frames, we often simply say that $\Box p \to \Box\Box p$ defines transitivity. It is also usual to say that $\Box p \to \Box\Box p$ corresponds (at the level of frames) to $\forall xyz(Rxy \wedge Ryz \to Rxz)$, or that $\forall xyz(Rxy \wedge Ryz \to Rxz)$ is a frame correspondent for $\Box p \to \Box\Box p$.

Now for an important question: how do we go about showing that a class of frames *cannot* be modally defined? Answering such questions is typically more demanding than proving the type of result noted in Proposition 27, for instead of checking that a given formula defines a given frame class, we now have to prove that no modal formula is capable of this. How can we prove such general results? By finding ways of transforming frames that preserve frame validity. For if we can show that a class of frames F is *not* closed under such a transformation, it follows that F is *not* modally definable. Let's take a closer look.

The first step is to find transformations that preserve frame validity. Three lie close to hand: the formation of disjoint unions, generated submodels, and bounded morphic images. In Section 3.2 we defined these constructions at the level of models, and they can be lifted to the level of frames simply by ignoring the requirements imposed on the valuations. For example, a bounded morphism between frames (W, R) and (W', R') is a function f from W to W' that satisfies the morphism condition (if Rwv, then $R'f(w)f(v)$) and the zag condition (if $R'f(w)v'$, then there exists a v such that $f(v) = v'$ and Rwv), and we say that frame (W', R') is a bounded morphic image of frame (W, R) if there is a surjective bounded morphism from (W, R) to (W', R'). Lifting these constructions to the level of frames immediately gives us three validity preservation results:

THEOREM 28. *For all basic modal formulas φ we have that:*

1. *Let $\{\mathfrak{F}_i \mid i \in I\}$ be a family of frames. Then if $\mathfrak{F}_i \models \varphi$ for every i in I, we have that $\biguplus \mathfrak{F}_i \models \varphi$ too. That is, frame validity is preserved under the formation of disjoint unions.*

2. *Let $\mathfrak{F}'$ be a generated subframe of $\mathfrak{F}$. Then if $\mathfrak{F} \models \varphi$, we have that $\mathfrak{F}' \models \varphi$ too. That is, frame validity is preserved under the formation of generated subframes.*

3. *Let $\mathfrak{F}$ and $\mathfrak{F}'$ be frames and f a surjective bounded morphism from $\mathfrak{F}$ to $\mathfrak{F}'$. Then if $\mathfrak{F} \models \varphi$, we have that $\mathfrak{F}' \models \varphi$ too. That is, frame validity is preserved under the formation of bounded morphic images.*

Proof. We prove the result for bounded morphisms; we show the contrapositive. Given frames $\mathfrak{F} = (W, R)$ and $\mathfrak{F}' = (W', R')$ such that $\mathfrak{F}'$ is a bounded morphic image of $\mathfrak{F}$ under f, suppose that $\mathfrak{F}' \not\models \varphi$. This means that for some valuation V' on $\mathfrak{F}'$ and some point $w' \in W'$ we have that $(\mathfrak{F}', V'), w' \not\models \varphi$. Let V be the valuation on $\mathfrak{F}$ defined by $V(p) = \{u \in W \mid f(u) \in V'(p)\}$, for all proposition symbols p. Furthermore, let w be any point such that $f(w) = w'$; there must be at least one such point as f is surjective. Then the model $(\mathfrak{F}', V')$ is a bounded morphic image of the model $(\mathfrak{F}, V)$, and hence $(\mathfrak{F}, V), w \not\models \varphi$. ❑

Applying this theorem immediately gives rise to a crop of non-definability results. Here are some simple ones. Basic modal languages cannot define the class of simply connected frames, that is, the class of frames such that $\forall xy(Rxy \vee Ryx)$. Why not? Because this class is not closed under the formation of disjoint unions: taking the disjoint union of two frames with this property clearly results in a frame without it. As a second example, the basic modal languages cannot define the class of frames containing an isolated reflexive point. Why not? Because this class is not closed under the formation of generated subframes. For consider a frame consisting of two isolated points, one reflexive, the other irreflexive. This frame belongs to the required class, however the subframe generated by the irreflexive point does not. As a third example, the class of irreflexive frames is not modally definable. Why not? Because it is not closed under the formation of bounded morphic images (recall the bounded morphism of Figure 13 which collapses the natural numbers to a single reflexive point). But frame validity is preserved under this transformation, hence no modal formula can define irreflexivity. For more sophisticated applications of these validity preservation results, see van Benthem [137].

These results also give us insight into the semantic ideas behind Theorem 25. For consider a consistent normal logic. Suppose one of the frames on which it is valid contains an isolated irreflexive point; then (appealing to the preservation of validity under generated subframes) the frame consisting of just that single point validates the logic too. So suppose that no frame containing an isolated point validates the logic. But this means that in all frames that validate the logic, every point has at least one successor. But if we map all the points in such a frame to a singleton reflexive point, the mapping is a bounded morphism. Hence it follows that the logic is validated on frames consisting of isolated reflexive points.

As we shall soon see, the three frame transformations just introduced all play a role in the Goldblatt-Thomason Theorem, a characterisation of modally definable classes of elementary frames. But a fourth transformation, namely the formation of *ultrafilter extensions*, is also needed to complete the statement of this celebrated result, so let's take this opportunity to define this (somewhat more complex) frame construction. First we recall a standard mathematical concept. Given a non-empty set W, a *filter* F over W is any subset of 2^W (the power set of W) that contains W and is closed under finite intersection (that is, if $X, Y \in F$ then $X \cap Y \in F$) and set-theoretic inclusion (that is, if $X \in F$ and $X \subseteq Y \subseteq W$ then $Y \in F$). A filter is called *proper* if it is distinct from 2^W. An *ultrafilter* is a proper filter U such that for all $X \in 2^W$, $X \notin U$ iff $(W \backslash X) \in U$. A standard result assures us that any proper filter can be extended to an ultrafilter. Bearing this in mind, we make the following definition:

DEFINITION 29 (Ultrafilter Extensions of Frames). Let $\mathfrak{F} = (W, R)$ be a frame. For any $X \subseteq W$ we define $l(X)$ to be $\{w \in W \mid \text{for all } v \in W, \text{if } Rwv \text{ then } v \in X\}$. Then the ultrafilter extension $ue(\mathfrak{F})$ of $\mathfrak{F}$ is defined to be the frame $(uf(W), R^{ue})$, where $uf(W)$ is the set of all ultrafilters on W and R^{ue} is the relation consisting of all pairs of ultrafilters U, U' such that for all $X \subseteq W$, if $l(X) \in U$, then $X \in U'$.

We can now state the required theorem. Note that the direction of validity preservation is

the reverse of that found in Theorem 28. That is, here frame validity is preserved from the transformed frame (here the ultrafilter extension) back to the original one:

THEOREM 30. *For any basic modal formula φ, if $ue(\mathfrak{F}) \models \varphi$ then $\mathfrak{F} \models \varphi$ does too. That is, frame validity reflects ultrafilter extensions.*

Proof. The use of ultrafilter extensions in modal logic traces back to Goldblatt [57, 58], van Benthem [130], and Fine [44]. For a detailed proof of this theorem, see Proposition 2.59 and Corollary 3.16 of Blackburn, de Rijke and Venema [13]. ❑

Although this transformation is harder to visualise than the previous three, it too gives rise to some simple non-definability results. Here's a nice example, taken from Goldblatt and Thomason [60], showing that the class of frames satisfying $\forall x \exists y(Rxy \wedge Ryy)$ is not modally definable. We can see this as follows. The ultrafilter extension of $(\mathbb{N}, <)$, the natural numbers in their usual order, looks a bit like a gigantic lolly-pop. It has an infinite handle, an isomorphic copy of $(\mathbb{N}, <)$, consisting of all the principal ultrafilters (that is, those ultrafilters which contain a singleton set $\{n\}$, where n is a natural number). This is followed by the lolly: an uncountable collection of non-principal ultrafilters which are all related to one another and reflexively related to themselves. Hence $ue(\mathbb{N}, <)$ has the property $\forall x \exists y(Rxy \wedge Ryy)$. Why? Because every point in the frame is related to the reflexive points in the lolly. However this formula is clearly not valid on the original frame $(\mathbb{N}, <)$. As frame validity reflects ultrafilter extensions, it follows that the class of frames satisfying $\forall x \exists y(Rxy \wedge Ryy)$ is not modally definable. For further discussion of ultrafilter extensions from a model-theoretic perspective, see Chapter 5 of this handbook. There is also an important algebraic perspective on ultrafilter extensions, which is discussed in Chapter 6.

5.3 *Frame correspondence and second-order logic*

Now that we have some idea of what basic modal languages can (and cannot) say about frames, we turn to the second question: how do they say it? And here we encounter something interesting. Note that all four classes of frames mentioned in Proposition 27 are definable by simple first-order formulas — and this is actually rather puzzling. After all, if you think about what it means for a basic modal formula $\varphi(p_1, \ldots, p_n)$ to be valid on a frame, we see that this concept is essentially *second-order*: we quantify across all possible valuations, and valuations assign *subsets* of frames to proposition symbols.

We can make this second-order perspective precise with the help of the standard translation. Let $\mathfrak{F}$ be a frame, let $\mathfrak{M} = (\mathfrak{F}, V)$ be any model over $\mathfrak{F}$, and let w be any point in $\mathfrak{F}$. By Proposition 3 we have that

$$(\mathfrak{F}, V), w \models \varphi(p_1, \ldots, p_n) \text{ iff } (\mathfrak{F}, V) \models \mathrm{ST}_x(\varphi)(P_1, \ldots, P_n)[x \leftarrow w].$$

(Here $P_1, \ldots, P_n$ are the monadic predicate symbols used to translate the proposition symbols $p_1, \ldots, p_n$.) How do we lift this equivalence (which lives at the level of models) to an equivalence at the level of frames (the level where validity is the primary semantic concept)? Very straightforwardly. A formula is valid on a frame iff it is satisfied at any point in the frame under any assignment of subsets of the frame to the proposition symbols. So we only need to universal quantify over the points that can be assigned to x (a first-order quantification) and over the assignments to the monadic symbols $P_1, \ldots, P_n$ (a second-order quantification). Doing so gives us the fundamental correspondence between frame validity and second-order logic:

$$\mathfrak{F} \models \varphi(p_1, \ldots, p_n) \text{ iff } \mathfrak{F} \models \forall P_1 \cdots P_n \forall x \mathrm{ST}_x(\varphi).$$

In short, frame validity systematically treats modal formulas φ as the universal monadic second-order closure of their standard first-order translations on relational models. The second-order upgrade of the first-order correspondence language is often called the *frame correspondence language* or the *second-order correspondence language*.

Let's look at an example. Recall that in Section 2.2 we showed that the standard translation of $p \to \Diamond p$ was $Px \to \exists y(Rxy \wedge Py)$. So if we ask what $p \to \Diamond p$ defines at the level of frames we can give an immediate answer: it defines the class of frames satisfying the following monadic second-order formula:

$$\forall P \forall x(Px \to \exists y(Rxy \wedge Py)).$$

Now, it's certainly pleasant to be able to systematically calculate frame correspondences for modal formulas in this way — but the puzzle remains. Indeed, if anything it has become more acute. For most of the modal formulas encountered in practice correspond to simple first-order conditions on frames, yet these conditions are systematically expressed using rather complex second-order expressions. The translation just considered is a good example. As the reader should check, $p \to \Diamond p$ corresponds to the first-order formula $\forall x Rxx$ (that is, it defines reflexivity). And if you think about it, you will see that $\forall P \forall x(Px \to \exists y(Rxy \wedge Py))$ is indeed a rather roundabout way of expressing reflexivity. For a start, it's easy to see that this sentence is true on any reflexive frame. Conversely, if this sentence is true on a frame (W, R), then $Px \to \exists y(Rxy \wedge Py)$ must be true under any assignment to the free variables x and P. Hence, for any $w \in W$, this formula is true if we assign w to x and $\{w\}$ to P. This assignment makes the antecedent true (indeed, it is the *minimal* valuation required to make the antecedent true; the significance of this remark will become clear when we discuss the Sahlqvist Correspondence Theorem) so we must have that $\exists y(Rxy \wedge Py)$ is true too. But this is only possible if Rww. Hence, as w was arbitrary, this means that R must be reflexive, and thus the original second-order sentence really does express reflexivity. As we said earlier, one of the key questions we are interested in is *how* modal languages talk about frames. And now we have an answer. They do so via a detour through second-order logic.

Moreover, this detour is *not* eliminable. That is, while experience shows that most common modal formulas correspond to first-order conditions on frames, some modal formulas define conditions that are *not* elementary. A famous case is Löb's formula, $\Box(\Box p \to p) \to \Box p$, the characteristic axiom of the logic **GL**. This defines the conjunction of the transitivity of R with the converse well-foundedness of R (that is, it forbids the existence of infinite chains of related points $w_1 R w_2 R w_3 R w_4 R w_5 \ldots$). This condition is non-elementary, as an appeal to the Compactness Theorem for first-order logic shows. Another well-known modal axiom that defines a non-elementary class of frames is the McKinsey formula $\Box\Diamond p \to \Diamond\Box p$. This can be shown by appealing to the Löwenheim-Skolem Theorem for first-order logic. For full proof details for both the Löb and McKinsey examples, see Blackburn, de Rijke and Venema [13].

Summing up, we are confronted with an intriguing situation. At the level of frames, modal formulas systematically correspond to second-order conditions on frames. Nonetheless, in many common cases these second-order conditions turn out to be equivalent to first-order conditions. This raises some interesting questions. Are there criteria that demarcate modal formulas that are essentially first-order at the level of frames from the genuinely second-order ones? And can we characterise the elementary frame classes that are modally definable?

5.4 First-order frame definability

As we have just learned, the link between first-order definable frame classes and modal logic is not straightforward. Nonetheless, some elegant general results are known, and we shall briefly discuss three of them here. We first note two results which bear upon the demarcation issue: the Sahlqvist Correspondence Theorem (which isolates a large class of formulas all of which define elementary classes of frames) and a model-theoretic characterisation of the modal formulas which define elementary frame classes. Following this we discuss the celebrated Goldblatt-Thomason Theorem, a model-theoretic characterisation of the elementary frame classes that are basic modal definable. All three results (and others bearing on the theme of elementary frame definability) are discussed in greater detail in Chapter 5 of this handbook.

Let's start with the Sahlqvist [111] result. Upon closer inspection, first-order frame conditions often arise because of the syntactic shape of the defining modal formula — for example the quantifier shape of the first-order formula for transitivity is matched by the sequence of boxes in $\Box p \to \Box\Box p$. The following theorem gives us a natural account of such correspondences. It trades systematically on the idea (noted when we discussed the second-order definition of reflexivity) of substituting minimal verifying valuations in antecedents.

THEOREM 31 (Sahlqvist Correspondence Theorem). *There is an effective method for computing first-order equivalents for Sahlqvist formulas, that is, formulas of the form $\varphi \to \psi$ with antecedents φ constructed from atoms (possibly prefixed by boxes) using conjunctions, disjunctions and diamonds, while consequents ψ can be any modal formula with only positive occurrences of proposition symbols.*

Proof. The effective method (in the form originally introduced by van Benthem [128, 131]) is usually called the substitution algorithm. The following example will give an idea of how it works. The 4 formula, $\Box p \to \Box\Box p$, is a Sahlqvist formula and its second-order translation is

$$\forall P \forall x(\forall y(Rxy \to Py) \to \forall y(Rxy \to \forall z(Ryz \to Pz))).$$

Now, if we could eliminate all the occurrences of P in this formula, we would render the second-order quantification needed to express validity vacuous. But can P be eliminated in a semantically sensible way? Because of the syntactic restrictions that Sahlqvist formulas conform to, it turns out that it can. We do so by replacing P by a first-order expression describing the *minimal* valuation needed to make the antecedent of $\Box p \to \Box\Box p$ true. Now, the minimal way of making $\Box p$ true is to make p true at all successors of the point of evaluation x, so the required substitution is $Pu := Rxu$. Performing this substitution yields the following first-order expression:

$$\forall x(\forall y(Rxy \to Rxy) \to \forall y(Rxy \to \forall z(Ryz \to Rxz))).$$

The antecedent is now tautologically true, and dropping it leaves us with the expression

$$\forall x \forall y(Rxy \to \forall z(Ryz \to Rxz)).$$

But this is a first-order formula expressing transitivity. For a precise specification of the substitution algorithm, and a proof that it works as required, see Blackburn, de Rijke and Venema [13]. The heart of the proof is to show that a Sahlqvist antecedent is true under any value for its proposition symbols iff it is true under its *minimal* values. ❑

The Sahlqvist Correspondence Theorem and its proof method are very powerful and can be extended to far stronger modal languages. Nevertheless there are also modal formulas which

express first-order conditions on frames that are not covered by the theorem. The **K4.1** axiom

$$(\Box p \to \Box\Box p) \land (\Box\Diamond p \to \Diamond\Box p)$$

is a conjunction of the 4 axiom with the McKinsey axiom. It defines the class of frames with a transitive and atomic relation, that is the class of transitive frames such that $\forall x \exists y (Rxy \land \forall z (Ryz \to z = y))$. But this first-order equivalence cannot be computed using the substitution method. See van Benthem [137] or Blackburn, de Rijke and Venema [13] for further discussion.

So the Sahlqvist result doesn't fully pin down the modal formulas that define elementary frame classes. However model-theoretic characterisations exist. For example we have:

THEOREM 32. *A modal formula defines a first-order frame property iff it is preserved under taking ultrapowers of frames.*

Proof. For the original proof, see van Benthem [131]. For an introduction to ultrapowers, consult Chang and Keisler [23]. ❑

Closure under ultrapowers is an abstract feature, and it is not easy to use it to recognise whether a given modal formula is first-order over frames. But then no simple method can be expected to work: Chagrova [21] shows that the problem of determining whether a modal formula expresses a first-order condition on frames is undecidable.

But now for our other question: which elementary classes of frames are modally definable? The classic result here is the Goldblatt-Thomason Theorem. This tells us that the four frame preservation results noted earlier are not merely necessary, they are also *sufficient* to characterise first-order frame definability:

THEOREM 33 (Goldblatt-Thomason Theorem). *A first-order frame property is modally definable iff it is preserved under taking disjoint unions, generated subframes, bounded morphic images, and reflects ultrafilter extensions.*

Proof. The left-to-right direction is just a restatement of the results noted in Theorems 28 and 30. The real work lies in the converse. The original proof, due to Goldblatt and Thomason [60] was algebraic; we briefly discuss this approach in Section 7.1, and an algebraic proof is given in Chapter 6 of this handbook. Nowadays there are also purely model-theoretic proofs; see van Benthem [133] for the earliest of these. ❑

5.5 Correspondence in richer languages

Throughout this section we have kept our eyes firmly on the goal of understanding modal expressivity with respect to elementary frame classes. This is an important topic (after all, we want to understand as much as possible about the route modal logic over frames takes from monadic second-order logic back to first-order logic) but it is also natural to wonder about the expressivity of modal logic with respect to non-elementary frame classes. Unfortunately, it is harder to come up with elegant answers here. In particular, we can't expect sweeping model-theoretic characterisations. Model-theoretic characterisations of elementary frame definability, such as Theorem 32 and the Goldblatt-Thomason Theorem, rest on the conceptual edifice of first-order model theory. Second-order model theory is nowhere near as well developed.

Nonetheless, some interesting results are known. For example, it turns out that we can apply the ideas underlying the proof of the Sahlqvist Correspondence Theorem beyond the confines of first-order logic. Let's briefly consider what is involved. The following discussion is based on

van Benthem [138]. Chapter 5 of this handbook contains a more detailed discussion of related material.

The substitution algorithm for Sahlqvist formulas runs into difficulties with more complex antecedents; a classic example is Löb's formula $\Box(\Box p \to p) \to \Box p$, which defines a non-elementary class of frames. But let's reflect on *why* we compute the minimal antecedent values for Sahlqvist formulas. In fact there are two reasons. Firstly, because Sahlqvist antecedents are true under any value for their proposition symbols iff they are true under their *minimal* values. Secondly, because such minimal predicates are first-order definable. Now, as it happens, the Löb antecedent does not fulfil the first-order definability criterion, but this does not mean that all that can be said is that the Löb's formula is intrinsically second-order — for, as it turns out, there is a smallest semantic value for the predicate P which will make its antecedent true. This is the set of points in the frame obtained by taking the *intersection* of all predicates P validating $\Box(\Box p \to p)$ where p is interpreted as P. Such a set must exist, because the standard translation of the Löb antecedent has a special syntactic form. Call a first-order formula $\varphi(P)$ *intersective* if it has one of the forms:

1. $\forall x(\psi(P, Q, x) \to Px)$, with P occurring only positively in $\psi(P, Q, x)$.
2. $\psi(P, Q)$, with P occurring only negatively in ψ.

It is easy to show that all formulas $\varphi(P)$ of this form have the above-mentioned *intersection property*: if $\varphi(P)$ holds for any predicate P it holds for the intersection of all predicates P satisfying it.

Thus it makes sense to talk about $minP.\varphi(P)$, the *minimal* satisfying predicate. Of course, such predicates need not be first-order definable, but it is not hard to show that minimal predicates for intersective first-order formulas are definable in a well-known extension of first-order logic, namely *LFP(FOL)*, first-order logic with monotonic fixed-points (we shall introduce the idea of monotonic fixed-points in more detail when we discuss the modal μ-calculus in Section 6). *LFP(FOL)* has many uses in computer science; it lies between first-order and second-order logic, and retains many useful model-theoretic properties such as invariance for potential isomorphism (see Ebbinghaus and Flum [35] for an introduction to *LFP(FOL)*).

Now, once we have such a minimal value for the antecedent predicates, it can be substituted into the consequent to obtain a frame equivalent just as before — though now, of course, we obtain an equivalent in *LFP(FOL)*. To return to our example, the standard translation of the Löb antecedent $\forall y((Rxy \wedge \forall z(Ryz \to Pz)) \to Py)$ is indeed intersective in the above sense. Therefore, the corresponding frame property of the Löb formula can be computed and (as we would expect) the result is an *LFP(FOL)* formula defining the property of transitivity plus converse well-foundedness. As a second example, consider the axiom of cyclic return:

$$(\Diamond p \wedge \Box(p \to \Box p)) \to p.$$

Again, this is not a Sahlqvist formula. But again, the antecedent is intersective (once we have moved out the modal $\Diamond$ to become a prefixed universal quantifier, as before in the substitution algorithm) and gives rise to a simple fixed-point computation for an equivalent frame property:

> *Every point x with an R-successor y can be reached from y by a finite sequence of successive R-steps.*

We can express this condition in *LFP(FOL)* as follows. First we define the concept of transitive closure:

$$R^+xy \;=_{def}\; minS, xy.Rxy \vee \exists z(Rxz \wedge Szy).$$

We can then capture the stated frame condition by insisting that:

$$\forall xy(Rxy \rightarrow R^{+}yx).$$

This is the beginning of a further layering of modal formulas with respect to semantic complexity. For there are also modal formulas with frame equivalents which cannot be expressed in *LFP(FOL)*. One example is the well known axiom in tense logic expressing Dedekind Completeness of linear orders, which is not preserved under the potential isomorphism between the rationals and the reals. And recently, van Benthem and Goranko have shown that the McKinsey formula, whose antecedent is typically non-intersective, does not correspond to any *LFP(FOL)* formula.

We started this chapter by saying that the process interpretation is a fundamental way of viewing modal logic. The present discussion shows that there is a natural link between modal logic and a far more sophisticated logic of processes, namely *LFP(FOL)*. We will return to the process interpretation in Section 6 when we examine Propositional Dynamic Logic and the modal μ-calculus, stronger modal languages which, like *LFP(FOL)*, can express some non-elementary concepts, such as transitive closure.

5.6 *Remarks on computability*

In Section 4 we contrasted the PSPACE decidability of modal logic with the undecidability of first-order logic. But these results concerned satisfiability and validity on the class of all frames. Suppose we restrict attention to particular classes of frames defined by basic modal formulas. There is no reason to suppose that modal satisfiability and validity problems over such frame classes will always be in PSPACE, or even that they will be decidable. And indeed, in many cases they are not not.

In some cases, restricting attention to a certain class of frames may lower the computational complexity. For example, suppose we restrict attention to the frames defined by $\Diamond p \rightarrow \Box p$, that is, the class of frames in which R is a partial function. Then the task of testing basic modal formulas for satisfiability becomes NP-complete, that is, no worse than the satisfiability problem for propositional logic. This is because (as the reader can easily check) if a basic modal formula φ has a model based on a frame in this class, then it not only has a finite model in this class, but a model containing at most $n + 1$ points, where n is the number of modalities in φ. Thus a non-deterministic algorithm which guesses a model, checks that it belongs to the frame class, and verifies that the formula is satisfied on it, runs in time polynomial in the size of φ.

But restricting attention to particular frame classes can easily result in undecidable problems. A recurring theme is the distinction between tree-like and grid-like models. We have already discussed why tree-like models are relevant to modal decidability over the class of all models; here we'll merely add that many more modal decidability results can be proved by appealing to *Rabin's Theorem* (see [107]), which in its simplest form shows that the monadic second-order theory of binary branching trees is decidable. Grid-like models, on the other hand, are (roughly speaking) those that contain regions that look like $\mathbb{N} \times \mathbb{N}$ (the product of the natural numbers with itself) under two orderings: the horizontal ordering (that is, $(j, k)R^{h}(j + 1, k)$), and the vertical ordering (that is, $(j, k)R^{v}(j, k+1)$) which together give rise to the characteristic grid-like shape. Now, it is hard to give precise generalisations, but experience shows that while even very strong modal languages tend to be decidable over tree-like models, even quite weak languages can be undecidable over grid-like models; we shall note such an example in Section 6 when we discuss combinations of modal logics. Such undecidability results ultimately trace back to the possibility

of encoding the $\mathbb{N} \times \mathbb{N}$ *tiling problem*, which is known to be undecidable. For a detailed account of the tiling problem, and a proof that it is undecidable, see Berger [12]. Here we'll simply say that it is essentially a geometrical puzzle. We are presented with a finite collection of square tile types, of fixed orientation. Each edge of each tile type is coloured. The $\mathbb{N} \times \mathbb{N}$ tiling problem asks: is it possible to write an algorithm which, when presented with such a collection of tile types, can correctly determine whether or not $\mathbb{N} \times \mathbb{N}$ can be tiled, using only tiles of the given type, in such a way that colours on adjacent tile edges match? That is, is it possible to place a tile (of one of these types) on each point of $\mathbb{N} \times \mathbb{N}$, in such a way that colours match both vertically and horizontally? For some tile types, this is possible, for others it is impossible. However there is no algorithm for deciding for which tile types this can be done; it is a simple, and elegant, example of a computationally undecidable problem. Showing that a modal logic is strong enough to encode this problem is often a straightforward way of showing its undecidability; see Blackburn, de Rijke and Venema [13] for examples of how to use the tiling problem in this way.

In a slogan: trees tend to be safe, but beware of grids. Somewhat poetically, we can imagine modal logic as a small boat navigating somewhere on the border between decidability and undecidability, as Figure 18 shows.

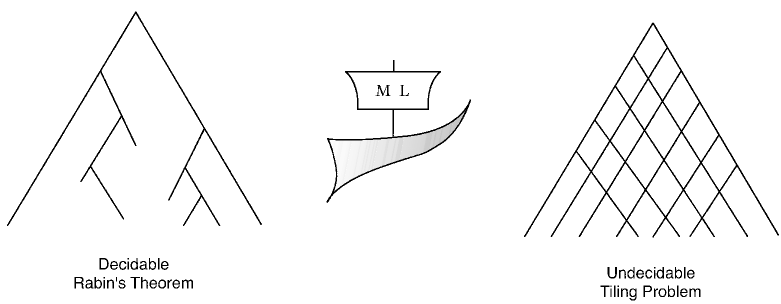


Figure 18. Modal logic: tacking between safety and danger.

Furthermore, it is important to realise that undecidable problems arise even when attention is restricted to finite frames; see, for example, Urquhart [127]. And indeed, even in the finite case, undecidability turns out to be the norm. It is straightforward to show that there are non-denumerably many distinct frame satisfiability problems over finite frame classes (an elegant demonstration of this, due to Spaan [118], is given as Exercise 6.2.4 of Blackburn, de Rijke and Venema [13]). As there are only denumerably many computable functions, undecidability is almost always guaranteed.

So what about recursive enumerability? That is, if we restrict attention to a class of frames F that is defined by a modal formula, is the theory of this frame class (that is, the set of formulas φ valid on all frames F) recursively enumerable? Well, if F is elementary, the answer is yes:

PROPOSITION 34. *Suppose that F is an elementary class of frames defined by a basic modal formula φ. Then the set of basic modal formulas that are valid on all frames in F is recursively enumerable.*

Proof. As F is an elementary class that φ defines, φ corresponds to some first-order formula α. Now a basic modal formula ψ is valid on frames for φ iff its second-order translation

$\forall P_1 \cdots P_n \forall x \text{ST}_x(\psi)$ is true in all models of the first-order formula α, that is, iff

$$\alpha \models \forall P_1 \cdots P_n \forall x \text{ST}_x(\psi),$$

where $\models$ is classical entailment. But as α is first-order, referring to R only, the predicates $P_1 \cdots P_n$ do not occur in α and hence this is equivalent to

$$\alpha \models \forall x \text{ST}_x(\psi).$$

But this is a first-order entailment, and as such entailments are recursively enumerable the result follows. ❑

However once we move beyond the elementary frame classes, even recursive enumerability is lost. A key result here is Thomason's [126] reduction of the standard consequence relation for the second-order correspondence language to the *global frame consequence* relation for a basic modal language with one modality. A basic modal formula φ is a global frame consequence of Γ if for all frames $\mathfrak{F}$, if $\mathfrak{F} \models \Gamma$, then $\mathfrak{F} \models \varphi$. It follows that global frame consequence is not recursively enumerable. Indeed, it is even Σ^1_1-complete, which means it is as hard to decide as the existential second-order theory of the natural numbers under the less-than-or-equal ordering. To put it another way: this is an example of a *highly undecidable* problem. For further discussion of Thomasons's work in this area, see Chapter 7 of this handbook.

6 RICHER LANGUAGES

So far we've been dealing almost exclusively with the basic modal language. We've seen that the key to its expressive power lies in the notion of bisimulation and that (at least when interpreted over the class of all models) it has better computational properties than first-order logic. All in all, the basic modal language is really rather elegant, so we might be tempted to ask: is it possible to lift (at least some of) its attractive properties to stronger languages? That is, can we design richer modal languages that retain, or even enhance, those features that make the basic modal language special? In fact, modal logicians have been experimenting with richer languages for years, and in this section we survey some of their work. As we shall see, this line of work adds a new dimension to our understanding of modal logic and relational semantics.

But what should count as a richer modal language? It's easier to explain what shouldn't. Here's an obvious example. It is straightforward to extend our basic definitions to cover *polyadic modalities* (that is, n-place diamonds and boxes). Simply work with models in which there is an $n+1$-place relation R^m for every n-place diamond $\langle m \rangle$. We interpret $\langle m \rangle$ using the following satisfaction clause:

$$\begin{array}{lll} \mathfrak{M}, w \models \langle m \rangle(\varphi_1, \ldots, \varphi_n) & \text{iff} & \text{for some } v_1, \ldots, v_n \in W \text{ such that } R^m w v_i \ldots v_n \\ & & \text{we have } \mathfrak{M}, v_1 \models \varphi_1 \text{ and } \ldots \text{ and } \mathfrak{M}, v_n \models \varphi_n. \end{array}$$

Now, such n-place modalities are undeniably useful for certain purposes, especially when interpreted over restricted classes of frames. For example, when working with spatio-temporal structures, we might want to add a three place modality to capture the notion of "between", or we might want to explore the logical theory of function composition, as is done in the branch of modal logic known as arrow logic (see Marx and Venema [94]). Nonetheless, when working with the class of all models, developing the basic semantic theory (standard translation, bisimulation,

and so on) of polyadic modal operators is essentially a matter of sprinkling our earlier work with additional indices.

As we shall see, the richer languages explored in this section offer much more. Moreover, their richness takes us in many different directions. Sometimes the enrichment consists of taking a standard language and insisting that a modality be interpreted by some mathematically fundamental relation (the universal modality is a good example). Sometimes the enrichment takes the form of more complex satisfaction definitions (both temporal logic with Until and Since and conditional logic are examples of this). In other cases, syntactic enhancements are introduced to support novel semantic capabilities (hybrid logic, propositional dynamic logic, and the modal μ-calculus all do this) and in one case (the guarded fragment) we enrich by abandoning modal syntax and using first-order syntax instead. Moreover, it is also possible to enrich by combining logics. For example, we might combine two propositional modal logics to enable some application domain to be more accurately modeled, or we might combine modal logic with first-order logic, a move which takes us to the historical heartland of philosophical applications of modal logic. As we shall see, modal logicians have been extremely creative when it comes to devising richer languages.

Of course, this variety raises a question of its own: what, if anything, do all these richer languages have in common? That is, what makes them all modal? This is not an easy question to answer. Nonetheless, as we work our way through this landscape a number of themes will recur: robust decidability, the importance of bisimulations, and characterisations of fragments of first- and second-order logic. As we shall see at the end of the section, the idea of restricted quantification that underlies the guarded fragment goes a long way towards accounting for these properties, for both first- and second-order enrichments. Moreover, it is possible to draw on ideas from abstract model theory and prove Lindström-style characterisation results. In short, we will often be able to lift much of the fundamental semantic theory for basic modal logic to a whole new level, a good indication that the enrichments discussed below are, in an important sense, genuinely modal.

6.1 *The universal modality*

Time to feed the bears again. As we said in Section 4, some problems demand a global perspective. We sometimes want to view a modal formula as a general background constraint, something that must be satisfied at *all* points in a model. Indeed, because of the importance of background constraints, in many practical situations we are primarily interested in the local-global satisfiability problem, which we formulated as follows: given basic modal formulas φ and ψ, is there a model which locally satisfies φ and globally satisfies ψ? Now, description logic, with its two level architecture of TBoxes (which impose general constraints) and ABoxes (which give information about particular individuals), acknowledges the importance of this problem (the information in a TBox has to be globally satisfied, while the information in an ABox only has to be locally satisfied). But the ability to impose global constraints is not incorporated into description logic concept languages (which are essentially notational variants of the basic modal languages we are familiar with) and this raises an interesting question. Is it possible to internalise the notion of global satisfiability in a modal language? And if so, what happens?

Let's introduce the *universal modality* and find out. To keep things simple, suppose we are working in a language with just one modality. We shall add a second modality, and will write E for its diamond form, and A for its box form. The interpretation of E and A is fixed: in any model $\mathfrak{M} = (W, R, V)$, both modalities must be interpreted using the universal relation $W \times W$. That

is, the satisfaction definition for these modalities is:

$$\begin{array}{lll} \mathfrak{M}, w \models E\varphi & \text{iff} & \text{there is a } u \in W \text{ such that } \mathfrak{M}, u \models \varphi \\ \mathfrak{M}, w \models A\varphi & \text{iff} & \text{for all } u \in W \text{ we have } \mathfrak{M}, u \models \varphi. \end{array}$$

Thus $E\varphi$ scans the entire model for a point that satisfies φ, while $A\varphi$ asserts that φ holds everywhere. We have imported the meta-theoretic notion of global truth into our modal object language, or to put it another way, we have internalised the TBox. Accordingly, we call E the *universal diamond*, and A the *universal box*. If it is irrelevant whether we mean E or its dual, we simply talk of the *universal modality*.

How can we be sure that adding the universal modality really increases the expressive power at our disposal? That is, are we certain that E and A are not already definable in the basic modal language? We are. One way to see this is via a bisimulation argument (see Example 2.4 in Blackburn, de Rijke and Venema [13] for such a proof). But an easy complexity-theoretic argument also establishes this. Let φ and ψ be basic modal formulas. Then the formula $A\psi$ expresses the global satisfiability problem (for the basic modal language) in our new language, and the formula $\varphi \wedge A\psi$ expresses the local-global satisfiability problem (for the basic modal language) again in our new language. Now, we remarked in Section 4 that both these problems are EXPTIME-complete. However the satisfiability problem for the basic modal language is PSPACE-complete. Hence (assuming that PSPACE is strictly contained in EXPTIME , the standard assumption) our ability to express these problems in the enriched language shows that the apparent increase in expressive power is genuine.

This in turn raises a new question. Because it can encode these problems, the satisfiability problem for the enriched language is at least EXPTIME-hard. But are some problem-instances even harder? No. Everything is solvable in EXPTIME.

THEOREM 35. *The satisfiability problem for the basic modal language enriched with the universal modality is EXPTIME-complete.*

Proof. See Hemaspaandra [65], or her earlier PhD thesis Spaan [118]. ❑

But the universal modality not only gives us extra expressivity at the level of models, it also increases our ability to define new classes of frames. Moreover, an elegant variant of the Goldblatt-Thomason Theorem holds for the enriched language. We'll discuss this result shortly, but let's first consider two examples of newly definable frame classes.

The class of frames of cardinality less than or equal to some natural number n (that is, frames in which $|W| \leq n$) is not definable in the basic modal language. Why not? Because basic modal validity is closed under the formation of disjoint unions. Hence any basic modal formula φ which allegedly defined this frame class could easily be shown to fail: simply by sticking together enough frames we could validate φ on frames of cardinality greater than n.

But this condition *is* definable with the help of the universal modality:

$$\bigwedge_{i=1}^{n+1} Ep_i \rightarrow \bigvee_{i \neq j} E(p_i \wedge p_j).$$

As the reader can easily check, this formula is valid on any frame where $|W| \leq n$, and can be falsified on any larger frame (in essence, the formula encodes the pigeonhole principle for $n+1$ pigeons and n holes). It follows that validity in the enriched language is not preserved under the

formation of disjoint unions. This, of course, is as it should be. We want a genuine *universal* modality, not something that can be fooled by the addition of new components.

Here's a second example. The condition $\forall x \exists y\, Ryx$ (that is, every point has a predecessor) is not definable in basic modal logic. Why not? Because modal validity is preserved under the formation of generated subframes. Any basic modal formula which putatively defined this class would have to be valid on the frame $(\mathbb{N}, R)$, where Rnm iff $n > m$, the natural numbers under the reverse ordering. But (by preservation under generated subframes) it would then have to be valid on the subframe generated by any number n. But in any such subframe, n has no predecessor, hence the condition is not basic modal definable.

But it *is* definable with the help of the universal modality:

$$p \to E\Diamond p.$$

It is easy to check that this formula defines the required condition, hence it follows that validity in the enriched language is not preserved under generated subframes. Again, this is the way it should be. A genuinely universal modality will not let us throw away points: its purpose is to keep an eye on the entire frame. It should be intolerant of both additions (disjoint unions) and deletions (generated submodels).

And now for the promised result: when it comes to defining elementary frame classes, intolerance towards disjoint unions and generated submodels is precisely what distinguishes the enriched language from the basic modal language. The following result is the Goldblatt-Thomason Theorem for the basic modal language, with closure under disjoint unions and generated subframes stripped away:

THEOREM 36. *A first-order definable class of frames is definable in the basic modal language enriched with the universal modality iff it is closed under taking bounded morphic images, and reflects ultrafilter extensions.*

Proof. See Goranko and Passy [61]. ❑

Three comments. First, adding the universal modality also increases our ability to define non-elementary frame classes. For example, the class of frames where the converse of the accessibility relation R is well-founded (that is, where it is impossible to form infinite R-successorship chains) is not definable in basic modal logic. Löb's formula, $\Box(\Box p \to p) \to \Box p$ doesn't quite pin this condition down (recall that it defines the conjunction of transitivity and converse well foundedness). But the following Löb-like formula in the enriched language does:

$$A(\Box p \to p) \to p.$$

(This example is from Goranko and Passy [61], the key reference on the universal modality.) Second, it is straightforward to extend the definition of bisimulation so that it works for the basic modal language enriched with the universal modality; all that needs to be done is to insist that the bisimulation be *total*, that is, that every element in each model is related to at least one point in the other; see de Rijke [30] for a brief discussion. Third, the universal modality has a big brother, the *difference operator*. The diamond form of this operator is written D, and $D\varphi$ is satisfied at a point w in a model if and only if φ is satisfied at some *different* point v (that is, the difference operator is interpreted using the $\neq$ relation on W). The difference operator is strong enough to define the universal modality ($E\varphi$ is just $\varphi \lor D\varphi$) but D cannot be defined using E (we leave the proof as an exercise). The difference operator arises naturally in many settings and, like the universal modality, has a smooth meta-theory; see de Rijke [29] for more information.

6.2 Hybrid logic

Basic modal languages have an obvious expressive weakness: they cannot name points. We cannot say this happened *then*, or that some *particular* individual has some property, or that two distinct sequences of processes take us from the current state to the *same* state. For example, in Figure 4 we let the nodes represent particular individuals such as Terry and Judy — but the basic modal language doesn't let us pick out these individuals. First-order logic, of course, lets us do this. We use constants to name individuals of interest, and the equality symbol for reasoning about their identity. No analogous mechanisms exist in basic modal logic. The *basic hybrid language* is the result of adding them.

At the heart of hybrid logic lies a simple idea, first introduced by Arthur Prior [104, 105] in the 1960s: sort the proposition symbols, and use *formulas as terms*. Let's do this right away. Take a language of basic modal logic (with proposition symbols p, q, r, and so on) and add a second sort of proposition symbol. The new symbols are called *nominals*, and are typically written i, j, k, and l. Both types of proposition symbol can be freely combined to form more complex formulas in the usual way. And now for the key change: *insist that each nominal be true at exactly one point in any model*. That is, insist (for any valuation V and nominal i) that $V(i)$ be a singleton set. We call the unique point in $V(i)$ the *denotation* of i. A nominal 'names' its denotation by being true there and nowhere else.

This change is far from negligible: already we have a more expressive logic. Consider the following basic modal formula:

$$\Diamond(r \wedge p) \wedge \Diamond(r \wedge q) \rightarrow \Diamond(p \wedge q).$$

This formula can be falsified, as the p-witnessing and q-witnessing points given by the antecedent may be distinct. But now consider the following hybrid formula:

$$\Diamond(i \wedge p) \wedge \Diamond(i \wedge q) \rightarrow \Diamond(p \wedge q).$$

This is identical to the preceding formula, except that we have replaced the proposition symbol r by the nominal i. But the resulting formula is valid. For now we have extra information: the p-witnessing and q-witnessing successors both make i true, so they are true at the same point, namely the denotation of i.

The addition of nominals is the crucial step towards the basic hybrid language, but we need a second ingredient too: *satisfaction operators*. These are operators of the form $@_i$, where i is a nominal. The formula $@_i\varphi$ asserts that φ is satisfied at the (unique) point named by the nominal i. That is:

$$\mathfrak{M}, w \models @_i\varphi \quad \text{iff} \quad \mathfrak{M}, u \models \varphi, \ \text{where } u \text{ is the denotation of } i.$$

Syntactically, satisfaction operators are modalities. And they are semantically well behaved. For a start, all instances of the modal distribution schema are valid:

$$@_i(\varphi \rightarrow \psi) \rightarrow (@_i\varphi \rightarrow @_i\psi).$$

Moreover, satisfaction operators also admit the modal generalisation law: if φ is valid, then so is $@_i\varphi$ (for any choice of i). Hence satisfaction operators are normal modal operators. Moreover, they are self-dual modalities, for all instances of $@_i\varphi \leftrightarrow \neg @_i\neg\varphi$ are valid. So we are free to regard satisfaction operators as either boxes or diamonds.

But for present purposes, the most important point about satisfaction operators is that they give us a modal perspective on the equality relation. To see this, note that formulas like $@_ij$ are

well formed. What does this formula assert? It says that "at the denotation of i, the nominal j is satisfied", or to put it another way, "the point named i is identical to the point named j". Hence the following schemas are valid: $@_i i$ (reflexivity of equality), $@_i j \to @_j i$ (symmetry of equality), $@_i j \wedge @_j k \to @_i k$ (transitivity of equality), and $@_i \varphi \wedge @_i j \to @_j \varphi$ (replacement). As we hoped, a modal theory of equality is emerging.

We will shortly characterise this theory, but before doing so let's glance at what is happening at the level of frames. Here too there is an increase in expressivity. None of the four first-order definable frame conditions listed below can be defined in basic modal logic. But it is easy to check that each is defined by the hybrid formula written next to them:

$$\begin{array}{rll} \forall x \neg Rxx & i \to \neg\Diamond i & \text{(irreflexivity)} \\ \forall xy(Rxy \to \neg Ryx) & i \to \neg\Diamond\Diamond i & \text{(asymmetry)} \\ \forall xy(Rxy \wedge Ryx \to x = y) & i \to \Box(\Diamond i \to i) & \text{(antisymmetry)} \\ \forall xy(Rxy \vee x = y \vee Ryx) & @_j\Diamond i \vee @_j i \vee @_i\Diamond j & \text{(trichotomy).} \end{array}$$

And now for the main result. Hybridisation has given us some sort of modal theory of equality. But how much of the corresponding first-order theory have we captured? Of course, now when we talk about "corresponding first-order theory" we mean: theory in the first-order correspondence language *enriched with constants and the equality symbol*.

The first step towards an answer is to extend the standard translation to cover nominals and satisfaction operators. So enrich the first-order correspondence language with constants and the equality symbol; to keep the notation uncluttered, we'll re-use the nominals as first-order constants. Then add the following clauses to the standard translation:

$$\begin{array}{rcl} \mathrm{ST}_x(i) & = & (x = i) \\ \mathrm{ST}_x(@_i\varphi) & = & \mathrm{ST}_i(\varphi). \end{array}$$

That is, nominals i are translated into first-order constants i, and satisfaction operators are translated by substituting the relevant first-order constant for the free-variable x. Note that this translation returns first-order formulas with *at most* one free variable x, not exactly one. This is because a constant may be substituted for the free occurrence of x. For example, the hybrid formula $@_i i$ translates into the first-order *sentence* $i = i$.

The second step is to extend the notion of bisimulation given in Definition 5 to make it suitable for the basic hybrid language and for the constant-enriched first-order correspondence language:

DEFINITION 37 (Bisimulation-with-names). A bisimulation-with-names between models $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$ is a non-empty binary relation E between their domains (that is, $E \subseteq W \times W'$) such that whenever wEw' we have that:

Atomic harmony: w and w' satisfy the same proposition symbols, and the same nominals.

Zig: if Rwv, then there exists a point v' (in $\mathfrak{M}'$) such that vEv' and $R'w'v'$, and

Zag: if $R'w'v'$, then there exists a point v (in $\mathfrak{M}$) such that vEv' and Rwv.

Closure: All points named by nominals are related by E.

It is easy to check that all basic hybrid formulas are invariant under bisimulations-with-names; the proof is an easy extension of Lemma 9. More interestingly, such bisimulations also give rise to a Characterisation Theorem:

THEOREM 38 (Hybrid Characterisation Theorem). *The following are equivalent for all first-order formulas $\varphi(x)$ in at most one free variable x:*

1. *$\varphi(x)$ is invariant for bisimulation-with-names.*
2. *$\varphi(x)$ is equivalent to the standard translation of a basic hybrid formula.*

Proof. That clause 2 implies 1 is a more or less immediate. The hard direction is showing that clause 1 implies 2. The original proof can be found in Areces, Blackburn and Marx [6]. ❑

In short, basic hybrid logic is a simple notation for capturing *exactly* the bisimulation-invariant fragment of first-order logic with constants and equality, or to put it another way, basic hybridisation is a mechanism for equality reasoning in propositional modal logic. And it comes cheap. Up to a polynomial, the complexity of the resulting decision problem is no worse than for the basic modal language we started with:

THEOREM 39. *The satisfiability problem for the basic hybrid language over arbitrary models is PSPACE-complete.*

Proof. See Areces, Blackburn and Marx [6]. ❑

A number of stronger hybrid languages have also been explored. One of the most interesting extensions is to add $\downarrow$ (the *downarrow binder*). This binds occurrences of nominals within its scope to the point of evaluation. That is, to evaluate $\mathfrak{M}, w \models \downarrow i.\varphi$, we evaluate $\mathfrak{M}, w \models \varphi$ but with all occurrences of the nominal i that were bound by $\downarrow$ now interpreted as naming w (for details on how to make this informal explanation precise, see Chapter 14 of this Handbook). To put it another way, $\downarrow$ lets us create a name for *here*, and this immediately increases the expressive power at our disposal. For example, in any model $\mathfrak{M}$, the formula $\downarrow i.\neg\Diamond i$ is true at precisely the irreflexive points; as we noted earlier, no such formula exists in the basic modal language, and indeed, no such formula exists in the basic hybrid language either.

Moreover, $\downarrow$ interacts beautifully with @. Intuitively, $\downarrow$ stores new values for nominals, and @ allows us to retrieve them. As an example of this interaction, consider the following formula which is true in any model at points with at least two successors:

$$\downarrow i.\Diamond\downarrow j.@i\Diamond\neg j.$$

This formula first names the point of evaluation i, it then declares that i has a successor which it names j, and then (with the help of @) it jumps back to i to assert that i also has a successor distinct from j.

But this increased expressivity comes at a price: by introducing $\downarrow$ we have sailed over the border into undecidability. As we remarked earlier, the ability to create grid-like models is a useful warning sign of undecidability, and the smooth interaction between $\downarrow$ and @ makes it easy to create the unit squares required to build grids:

$$\downarrow i.\Diamond(\neg i \wedge \downarrow j.\Diamond(\neg i \wedge \neg j \wedge \downarrow k.@i\Diamond(\neg i \wedge \neg j \wedge \neg k \wedge \downarrow l.\Diamond k))).$$

If you work through this formula you will see that it demands the existence of four distinct points, which it calls i, j, k, and l, such that Rij, Rjk, Ril and Rlk. Note the characteristic use of the embedded $@_i$ to jump us back to the original point of evaluation i; this enables us to construct a second path from i to k that goes via point l. Of course, moving from this observation to a proof that it is possible to code the tiling problem takes more work, but it can be done, and the upshot is: adding $\downarrow$ has moved us up to an undecidable fragment of first-order logic.

But which fragment? The answer has two natural formulations. The first has the now-familiar form of a Characterisation Theorem: it turns out that adding downarrow has moved us up to precisely that fragment of first-order logic which is *invariant under generated submodels*. The second answer has a more syntactic flavour: we have moved up to the *bounded fragment* of first-order logic. The bounded fragment consist of all first-order formulas built up from atomic formulas using the booleans and bounded quantifications of the form $\exists y(R\tau y \wedge \varphi)$ and $\forall y(R\tau y \rightarrow \varphi)$, where τ is a term that does not contain y. The bounded fragment arises naturally in set theory (see Levy [89]) and arithmetic (see Buss [20]). In the mid-1960s, Feferman and Kreisel [41, 40] characterised the bounded fragment as the fragment of first-order logic invariant under generated submodels. It is intriguing that hybrid logic should have arrived at the same fragment by such a different route.

For full formulations and proofs of these results, see Areces, Blackburn and Marx [6]. For a detailed overview of hybrid logic, covering the results mentioned and much else besides, see Chapter 14 of this handbook.

6.3 Temporal logic with Until and Since operators

We turn now to another historically early enrichment: the addition of the binary U (Until) and S (Since) operators. These were introduced in the late 1960s by Hans Kamp [76], who added them to Arthur Prior's basic (F and P based) tense logic, and proved an elegant result: U and S are expressively complete with respect to Dedekind complete strict total orders (we discuss Kamp's result below). But, beautiful though this is, it is not what led to the present popularity of these operators. Rather, around 1980, Gabbay, Pnueli, Shelah and Stavi [53] observed that Until offers precisely what is required to state what computer scientists call *guarantee properties*, and this led to its widespread adoption for reasoning about programs. Given the number of researchers currently active in temporal logic for program verification, Until may well be the best known and most widely used modal operator of all: it plays a key role in LTL (Linear Time Temporal Logic), CTL (Computational Tree Logic), and CTL* (a highly expressive system that contains both LTL and CTL as sublogics). For an introduction to these logics, see Chapter 11 of this handbook, or Clarke, Grumberg and Peled [25].

Now, we briefly met the Until operator in Section 4 when we discussed model checking. There we defined it in terms of R^+ and R^*, the transitive and reflexive transitive closures of the underlying relation R used by the $\Diamond$ over tree-like models. Here we shall define Until and Since in their most general form:

$$
\begin{array}{rll}
\mathfrak{M}, w \models U(\varphi,\psi) & \text{iff} & \text{there is a } v \text{ such that } Rwv \text{ and } \mathfrak{M}, v \models \varphi, \\
 & & \text{and for all } u \text{ such that } Rwu \text{ and } Ruv \text{ we have } \mathfrak{M}, u \models \psi. \\
\mathfrak{M}, w \models S(\varphi,\psi) & \text{iff} & \text{there is a } v \text{ such that } Rvw \text{ and } \mathfrak{M}, v \models \varphi, \\
 & & \text{and for all } u \text{ such that } Rvu \text{ and } Ruw \text{ we have } \mathfrak{M}, u \models \psi.
\end{array}
$$

Putting this in words, Until asserts that there is *some* point in the future where φ holds, and that at *all* points between the point of evaluation and this future φ-witnessing point, ψ holds. Since functions in the same way, but towards the past. Note the $\exists\forall$ pattern of quantification in the satisfaction definitions. These operators are neither diamonds nor boxes; they are something new and (as we shall see) more powerful.

What can we say with them? For a start, they have all the power of ordinary diamonds: $U(\varphi,\top)$ has the same meaning as $\Diamond\varphi$. But now we can say more: these operators are tailor-made for stating guarantee properties, requirements of the form "*Some event will happen, and*

until that event takes place, a certain condition will hold". For if we represent the event by φ and the condition by ψ, then $U(\varphi, \psi)$ clearly captures what is required.

But how can we be sure that we can't state guarantee requirements in the basic modal language? A simple bisimulation argument demonstrates this. Consider the two models shown in Figure 19. The two models are clearly bisimilar (simply link both points in the right-hand model

Figure 19. Until is not definable in basic modal logic.

to the single point in the left-hand model; all proposition symbols are false at all points in both models, though this is irrelevant to the following argument). This means that the two models agree on the truth of all basic modal formulas at all points. But the models disagree on the value of $U(\top, \bot)$. This formula is false in the model on the left, but true at both points in the model of the right. We conclude that no basic modal formula can capture the effect of Until.

But this is a little too easy. Until is typically used for temporal reasoning tasks, and the two models just shown have little to recommend them as flows of time. But it turns out that Until cannot be defined even if we work with models with more structure. For a start, even if we restrict our attention to transitive models, Until is not basic modal definable. For consider the two models shown in Figure 20; we are interested in the transitive closure of the relation indicated by the arrows. These models are bisimilar (link w_0 and w_1 with w', link t_0 and t_1 with t', and so on). So suppose that there is some formula in the basic modal language that captures the effect of $U(p, q)$. Any such formula would be true in the left-hand model at points w_0 and w_1. For consider what happens at w_0 (the argument for w_1 is analogous). There is a point to its future (namely v_1) that makes p true and at all points lying in between (and there is only one, namely u) we have that q is satisfied. However any such formula would be *false* in the right-hand model at w', for here there are *two* points between w' and v' (namely u' and t') and t' does not satisfy q. As w' is bisimilar to w_0 and w_1, we conclude that no basic modal formula can capture the effect of Until. And this result can be strengthened. Even if we restrict ourselves to linear models, the basic modal language can't define Until, and it can't do so on the real numbers either (see Proposition 7.10 in Blackburn, de Rijke and Venema [13]).

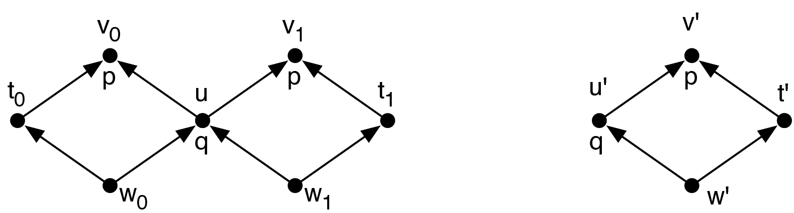


Figure 20. Even on transitive frames, Until is not definable in basic modal logic.

So adding S and U to the basic modal language yields new expressivity — but how much? We shall now discuss Kamp's Theorem, which shows that on certain classes of structures (a class

that includes the real numbers) these operators capture the entire one free variable fragment of the first-order correspondence language. This result was one of the earliest (and is still one of the most striking) purely semantic results in modal logic.

First, note that Until and Since correspond to fragments of the familiar first-order correspondence language that we have been working with throughout the chapter. After all, we can translate them as follows:

$$\begin{array}{rcl} \mathrm{ST}_x(U(\varphi,\psi)) & = & \exists z\,(Rxz \wedge \mathrm{ST}_z(\varphi) \wedge \forall y\,(Rxy \wedge Ryz \rightarrow \mathrm{ST}_y(\psi))) \\ \mathrm{ST}_x(S(\varphi,\psi)) & = & \exists z\,(Rzx \wedge \mathrm{ST}_z(\varphi) \wedge \forall y\,(Rzy \wedge Ryx \rightarrow \mathrm{ST}_y(\psi))). \end{array}$$

Incidentally, observe that we need three variables to specify this translation, whereas we only needed two for the basic modal language. Now, the three variable fragment of first-order logic is known to be undecidable, thus the translation doesn't give us an easy decidability result for the enriched modal language, though its satisfiability problem over arbitrary models is in fact decidable. We'll understand why a little later when we discuss the packed fragment.

So what does Kamp's Theorem say? First some preliminary definitions. Let M be a class of models. We say that a modal language is *expressively complete over* M, if every formula (in one free variable) from the first-order correspondence language is equivalent to a formula in the modal language (when we restrict attention to models from M). Which class of models is Kamp's Theorem about? A *strict total order* is any frame (with one binary relation R) that is transitive, irreflexive, and linear (that is, $\forall xy(Rxy \vee x = y \vee Ryx)$). A strict total order is *Dedekind complete* if every subset with an upper bound has a least upper bound. Standard examples of Dedekind complete strict total orders are the real numbers $(\mathbb{R}, <)$ and the natural numbers $(\mathbb{N}, <)$ under their usual orderings. And now we have:

THEOREM 40 (Kamp's Theorem). *The basic modal language enriched with U and S is expressively complete with respect to models based on Dedekind complete strict total orders.*

Proof. The original proof is in Kamp's thesis [76]. Elegant modern proofs, and proofs of stronger expressive completeness results, can be found in Gabbay, Hodkinson and Reynolds [52]. See also Chapter 11 of this handbook. ❑

Much more could be said about the Until and Since operators, but we will confine ourselves to the following remark. Because of their $\exists\forall$ pattern of quantification, for some time it was unclear how best to define a suitable notion of bisimulation. However Kurtonina and de Rijke [87] and Sturm [120] have given definitions which enable characterisation theorems to be proved.

6.4 Conditional logic

Although formulas of the form $\varphi \rightarrow \psi$ are often glossed as "if φ then ψ", the truth conditions that classical logic gives to the $\rightarrow$ symbol (and in particular, the fact that $\varphi \rightarrow \psi$ is true when φ is false) means that $\rightarrow$ does not mirror the more interesting meanings that conditionals can have in natural language. This has inspired numerous attempts to introduce conditional connectives (say, $>$) that better mimic the logic(s) of natural language conditionals. Indeed, such aspirations have given birth to an entire branch of logic, namely Relevance Logic, which nowadays is a well-established branch of the study of substructural logics (see Restall [108]).

But there is a modal approach to conditionals too. Its motivation comes from the following intuition: a conditional $\varphi > \psi$ can (often) be read as an *invitation* to assume the antecedent (perhaps making some adjustments to accommodate its truth) and check if the consequent is

true. A characteristic inferential feature of this reading is the failure of *monotonicity in the antecedent*. “If I catch the 6.22 train at Amsterdam Central (φ), I will be home on time (ψ)” is true on the most natural reading of the conditional, but adding an unusual further condition may make it false, as the sentence “If I catch the 6.22 train at Amsterdam Central (φ), and the dikes break (θ), I will be home on time (ψ)” demonstrates.

Models for modal-style conditional reasoning are triples $\mathfrak{M} = (W, C, V)$. Here W is a non-empty set (whose elements are usually called worlds), V is a valuation, and C is a ternary relation of *relative similarity* , or (as it is sometimes put in the literature) a relation of relative ‘comparison’ or ‘preference’ between worlds. It is useful to write $Cwuv$ as $C_w uv$ and to read this as saying that “w has more in common with u than v”. It is standard to demand that each C_w satisfies $\forall uvz(C_w uv \wedge C_w vz \rightarrow C_w uz)$, w-centred transitivity, and $\forall u C_w uu$, w-centred reflexivity. Moreover, some authors, most famously David Lewis, also demand w-centred comparability, that is, $\forall uv(C_w uv \vee C_w vu)$. A good way to visualise the relation $C_w uv$ is to think of u and v as two concentric circles around w. If u and v are distinct, then u is a concentric circle *closer* to w than v is.

The simplest truth condition for conditionals is the following, which come from David Lewis’s groundbreaking book “Counterfactuals” [90]. It fits in well with our intuitions (at least on finite models):

$\mathfrak{M}, w \models \varphi > \psi$ iff all minimal worlds in the w-centred ordering $C_w uv$ at which φ is true are also worlds where ψ holds.

This satisfaction clause can be phrased more succinctly as follows: all minimal φ-worlds are ψ-worlds.

Note that the φ-*minimal* worlds around w are the only ones we consider. As the minimal worlds satisfying the stronger condition $\varphi \wedge \theta$ need not be the ones satisfying φ, in this way we get a semantic distinction which accounts for the failure of monotonicity in the antecedent.

But what about *infinite* models? Then there need not be any minimal worlds satisfying the antecedent (we might have a chain of φ-satisfying concentric circles coming ever closer to w). Here’s a way of handling this: switch to the following more complex truth condition (to keep things readable, we shall write use $\varphi(v)$ as shorthand for $\mathfrak{M}, v \models \varphi$, and similarly for ψ):

$$\mathfrak{M}, w \models \varphi > \psi \quad \text{iff} \quad \forall u(\varphi(u) \Rightarrow \exists v(C_w vu \;\&\; \varphi(v) \;\&\; \forall z((C_w zv \;\&\; \varphi(z)) \Rightarrow \psi(z)).$$

This says that the conditional $\varphi > \psi$ holds if, whenever φ holds at some circle u, then there is some smaller circle v where φ holds such that all circles z within v satisfy ψ. This is rather awkward to process in first-order logic, but it can be clearly expressed in modal logic if we make use of a unary modality $\langle c \rangle$ (which looks inwards for a circle closer to the centre) together with the universal modality A. For then we can simply say:

$$\varphi > \psi \;=_{def}\; A(\varphi \rightarrow \langle c \rangle(\varphi \wedge [c](\varphi \rightarrow \psi)).$$

This more complex truth-condition validates a minimal logic which includes such principles as upward monotonicity in the consequent: $\varphi > \psi$ implies $\varphi > (\psi \vee \theta)$. Further properties of the similarity ordering enforce special axioms via standard frame correspondences. Assuming just reflexivity and transitivity yields the minimal conditional logic originally axiomatised by Burgess [19] and Veltman [143], while assuming also comparability of the ordering gives rise to the logics obtained by Davis Lewis.

What about complexity? A number of interesting results are known:

THEOREM 41. *The satisfiability problem for the minimal conditional logic (that is, where $C_w uv$ is transitive and reflexive) is PSPACE-complete when formulas with arbitrary nestings of conditionals are allowed, and NP-complete for formulas with bounded nesting of conditionals.*

Proof. See Friedman and Halpern [50]. These authors also prove that if uniformity is assumed (that is, if all worlds agree on what worlds are possible) the complexity rises to EXPTIME-complete, even for formulas with bounded nesting. Moreover, they show that if absoluteness is assumed (that is, all worlds agree on all conditional statements) the decision problem is NP-complete for formulas with arbitrary nesting. ❑

In general, conditional logic has not been studied semantically in the same style as most modal languages, though there is no reason why it cannot be. For example, bisimulations could be defined for $>$ in much the same spirit as they are defined for temporal logics with Until and Since. Likewise, issues of frame definability beyond the minimal setting can be explored; for example, van Benthem [137] notes correspondences between conditional axioms and triangle inequalities concerning concrete geometrical relations of relative nearness in space. Many recent technical developments in conditional logic, however, have to do with its connection with *belief revision theory* (see Gärdenfors and Rott [55]). In that setting, a conditional $\varphi > \psi$ means "if I revise my current beliefs with the information that φ, then ψ will be among my new beliefs"; see, for example, Ryan and Schobbens [110]. For more on these topics, see Chapters 20 and 21 of this handbook.

6.5 The guarded fragment

The richer modal languages so far examined have clearly been modal in a syntactic sense; all use the typical "apply operator to formula" syntax. The guarded fragment, however, arises as an attempt to *directly* isolate fragments of first-order logic that can plausibly be called modal. So the modal languages we shall consider here are syntactically first-order.

The clue leading to the guarded fragment is the standard translation of the modalities. This treats modalities as macros embodying *restricted* forms of first-order quantification, in particular, quantification restricted to successor states:

$$\begin{array}{rcl} \mathrm{ST}_x(\Diamond\varphi) & = & \exists y(Rxy \wedge \mathrm{ST}_y(\varphi)) \\ \mathrm{ST}_x(\Box\varphi) & = & \forall y(Rxy \rightarrow \mathrm{ST}_y(\varphi)). \end{array}$$

As we saw earlier, it is this restricted form of quantification that lets bisimulation emerge as the key model-theoretic notion. And bisimulation, via the tree model property, leads to decidability. Thus at least one pleasant property of modal logic can plausibly be traced back to its use of a restricted form of quantification. So it is natural to ask whether other first-order fragments defined by restricted quantification have such properties. This line of enquiry leads to the guarded fragment and its relatives.

The first step takes us to the guarded fragment, which was introduced by Andréka, van Benthem, and Németi [5]. Guarded formulas φ are built up as follows:

$$\varphi \quad ::= \quad Q\overline{x} \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \exists\overline{y}(G(\overline{x},\overline{y}) \wedge \varphi(\overline{x},\overline{y})) \mid \forall\overline{y}(G(\overline{x},\overline{y}) \rightarrow \varphi(\overline{x},\overline{y})).$$

Here $\overline{x}$ and $\overline{y}$ are finite tuples of variables, Q is a predicate symbol (of appropriate arity for the tuple $\overline{x}$), and G, the symbol used in the guard, is a predicate symbol too (thus the guard is an atomic formula). The key point to observe is that in the clauses for the quantifiers, all the free variables of φ appear in the guard. The set of all guarded first-order formulas is called the guarded fragment.

THEOREM 42. *The guarded fragment is decidable. Its satisfiability problem is 2EXPTIME-complete, and EXPTIME-complete if we have a fixed upper bound on the arity of predicates. Moreover, the guarded fragment has the finite model property.*

Proof. See Grädel [62] for the complexity results and a direct proof of the finite model property. An earlier (algebraic) proof of the finite model property can be found in Andréka, Hodkinson, and Németi [4]. ❑

The guarded fragment is a natural generalisation of the first-order formulas obtainable under the standard translation, but does it go far enough? For example, adding Until to a basic modal language yields a decidable logic, but the standard translation of $U(p, q)$, namely

$$\exists y\,(Rxy \wedge Py \wedge \forall z\,((Rxz \wedge Rzy) \rightarrow Qz)),$$

does not belong to the guarded fragment, and it can be shown that it is not equivalent to a formula in the guarded fragment either. This suggests that it may be possible to pin down richer restricted-quantification first-order fragments that retain decidability, and several closely related extensions of the guarded fragment, such as the loosely guarded fragment (see van Benthem [135]) and the packed fragment (see Marx [93]) have been proposed which do precisely this. Let's take a quick look at the packed fragment.

The packed fragment allows us to use *composite guards* γ instead of just atomic guards G. Let γ be a formula whose free variables are $\{x_1, \ldots, x_k\}$. Then γ *packs* $\{x_1, \ldots, x_k\}$ if γ is a conjunction of formulas of the form $x_i = x_j$, $R(x_{i_1}, \ldots, x_{i_n})$ or $\exists\overline{x}R(x_{i_1}, \cdots, x_{i_n})$, and moreover, for any two distinct free variables x_i and x_j, there is a conjunct in γ in which they both occur free. The packed fragment is the smallest fragment of modal logic that contains all atomic formulas, and is closed under boolean combinations and *packed quantification*. That is, if ψ is a packed formula, and γ packs ψ, and all the free variables of ψ are free in γ, then $\exists\overline{x}(\gamma \wedge \phi)$ and $\forall\overline{x}(\gamma \rightarrow \phi)$ are packed too.

As an example, consider again the standard translation of $U(p, q)$, namely

$$\exists y\,(Rxy \wedge Py \wedge \forall z\,((Rxz \wedge Rzy) \rightarrow Qz)).$$

This is not packed as the guard of the subformula $\forall z\,((Rxz \wedge Rzy) \rightarrow Qz))$ has no conjunct in which x and y occur together. But this is easy to fix. The following (logically equivalent) formula *is* packed:

$$\exists x\,(Rxy \wedge Py \wedge \forall z\,((Rxz \wedge Rzy \wedge Rxy) \rightarrow Qz)).$$

And indeed, the packed fragment turns out to be computationally well behaved:

THEOREM 43. *The packed fragment is decidable. Its satisfiability problem is 2EXPTIME-complete. Moreover, it has the finite model property.*

Proof. The complexity result follows from results in Grädel [62]. The original proof of the finite model property for the packed fragment (and the loosely guarded fragment) can be found in Hodkinson [68]; a more elegant proof can be found in Hodkinson and Otto [69]. ❑

In short, we have isolated two decidable fragments of first-order logic which are expressive enough to generalise many common modal languages. Moreover, these fragments have attractive properties besides decidability. Basic modal logic resembles first-order logic in most of its meta-properties, even those (such as Craig Interpolation, Beth definability, and the standard model-theoretic preservation theorems) that do not follow straightforwardly from the fact that it is a first-order fragment. The guarded fragment shares this good behaviour to some extent, witness the Łos-style preservation theorem for submodels given in Andréka, van Benthem, and Németi [5]. But subsequent work has shown that the picture is somewhat mixed. There is indeed a natural notion of guarded bisimulation (see Andréka, van Benthem, and Németi [5]) which characterises the guarded fragment as a fragment of first-order logic. Moreover, Beth definability holds (see Hoogland, Marx and Otto [71]). However Craig interpolation fails in its strong form, though it holds when we view guard predicates as part of the logical vocabulary (see Hoogland and Marx [70]).

This is a good moment to take stock of some of the first-order fragments we have encountered in the course of this chapter, and their interrelationships. Figure 21 summarises the relationships

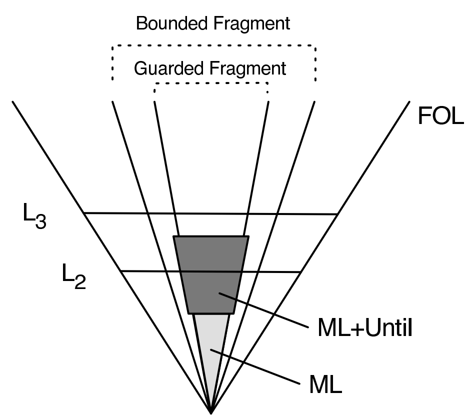


Figure 21. Some modally significant fragments of first-order logic.

between first-order logic, the more restricted (but undecidable) bounded fragment, and the still more restricted (but decidable) guarded fragment. Also shown are the fragments of first-order logic corresponding to the basic modal language, and the fragment corresponding to the basic language enriched with Until. Here L_2 and L_3 indicate the two and three variable fragments respectively; the basic language fits into the former, but the Until enriched language spills over into the latter.

6.6 *Propositional Dynamic Logic*

The richer modal languages so far discussed extend the first-order expressive power available for talking about models: the universal modality adds quantification over $W \times W$, hybridisation gives access to constants and equality, Until and Since and conditional logic add richer quantificational patterns, and the guarded-fragment cheerfully replaces modal syntax with first-order syntax. But the next two languages we shall discuss take us in a different direction: both add *second-order* expressive power. Now, in Section 5 we saw that modal languages have second-order expressive

power (via the concept of validity) at the level of *frames*. But in the languages we now consider, second-order expressivity arises directly: it is hardwired into the satisfaction definitions, and hence is available at the level of *models*. In particular, Propositional Dynamic Logic (henceforth PDL) offers us an (infinite collection of) transitive closure operators, and the modal μ-calculus offers us a general mechanism for forming fixed-points. Significantly, both PDL and the modal μ-calculus were born in theoretical computer science. Finite structures are crucial to the theory and practice of computation, and basic results of finite model theory (see Ebbinghaus and Flum [35]) show that first-order logic is badly behaved when interpreted over finite structures. Nowadays it is standard practice to extend first-order languages with second-order constructs (such as the ability to take transitive closures or form fixed-points) when working with finite models, and in the languages we now consider, such ideas are put to work in modal logic.

Let's start by looking at the weaker of the two languages, namely PDL. The underlying idea (to extend modal logic with a modality for every program) is due to Vaughan Pratt [102], and the language now called PDL was first investigated by Fisher and Ladner [47, 48]. PDL contains an infinite collection of diamonds. Each has the form $\langle\pi\rangle$, where π denotes a non-deterministic program. The intended interpretation of $\langle\pi\rangle\varphi$ is that "some terminating execution of π from the current state leads to a state with the information φ". The dual assertion $[\pi]\varphi$ states that "every terminating execution of π from the current state leads to a state with the information φ". Crucially, the inductive structure of programs is made explicit in PDL's syntax, as complex programs are built out of basic programs using four program constructors. Suppose we have fixed a set of basic programs a, b, c, and so on. We are allowed to define complex programs π over this base as follows:

Choice: if π_1 and π_2 are programs, then so is $\pi_1 \cup \pi_2$. It non-deterministically executes either π_1 or π_2.

Composition: if π_1 and π_2 are programs, then so is $\pi_1 ; \pi_2$. It first executes π_1 and then executes π_2.

Iteration: If π is a program, then so is π^*. It executes π a finite (possibly zero) number of times.

Test: if φ is a formula, then $\varphi?$ is a program. It tests whether φ holds, and if so, continues; if not, it fails.

Hence PDL makes available the following (inductively defined) algebra of diamonds. First we have diamonds $\langle a\rangle$, $\langle b\rangle$, $\langle c\rangle$, and so on, for working with the basic programs. Then, if $\langle\pi_1\rangle$ and $\langle\pi_2\rangle$ are diamonds and φ is a formula, $\langle\pi_1 \cup \pi_2\rangle$, $\langle\pi_1 ; \pi_2\rangle$, $\langle\pi_1^*\rangle$ and $\langle\varphi?\rangle$ are diamonds too. Note the unusual syntax of the test constructor diamond: it makes a modality out of a formula. This means that the sets of PDL formulas and modalities are defined by mutual induction.

How do we interpret PDL? Syntactically we're simply dealing with a basic modal language in which the modalities are indexed by a structured set. So a model for PDL will have the form we are used to, namely

$$(W, \{R^\pi \mid \pi \text{ is a program }\}, V),$$

a suitably indexed collection of relations together with a valuation. Moreover, the usual satisfaction definition is all that is required: diamonds existentially quantify over the relevant transitions, and boxes universally quantify over them. Nonetheless, something more needs to be said. Given the intended interpretation of PDL, most of these models are uninteresting. We want models built over frames which do justice to the intended meaning of our program constructors. Which models are these?

Nothing much needs to be said about the interpretation of the basic programs: any binary relation can be regarded as a transition relation for a non-deterministic program (though if we were interested in *deterministic* programs, we would insist on working with frames in which each basic program was interpreted by a partial function). Nor need much be said about the test operator. Unusual though its syntax is, its intended interpretation in any model $\mathfrak{M}$ is simply

$$R^{\varphi?} \;=\; \{(w,v) \mid w = v \text{ and } \mathfrak{M}, w \models \varphi\}.$$

But the three remaining constructors demand that we impose inductive structure on our frames. Here's what is required:

$$\begin{aligned} R^{\pi_1 \cup \pi_2} &= R^{\pi_1} \cup R^{\pi_2}, \\ R^{\pi_1 ; \pi_2} &= R^{\pi_1} \circ R^{\pi_2} \; (= \{(x,y) \mid \exists z\,(R^{\pi_1}xz \wedge R^{\pi_2}zy)\}), \\ R^{\pi_1^*} &= (R^{\pi_1})^*, \text{ the reflexive transitive closure of } R^{\pi_1}. \end{aligned}$$

These restrictions are the natural set-theoretic ways of capturing the "either-or" nature of non-deterministic choices (for $R^{\pi_1 \cup \pi_2}$), the idea of executing two programs in a sequence (for $R^{\pi_1;\pi_2}$) and the idea of iterating the execution of a program finitely many times (for $R^{\pi_1^*}$). Accordingly, we make the following definition. Let Π be the smallest set of programs containing the basic programs and the programs constructed over them using the constructors $\cup$, ;, and *. Then a *regular frame* over Π is a frame $(W, \{R^\pi \mid \pi \in \Pi\})$ where R^a is a binary relation for each basic program a, and for all complex programs π, R^π is the binary relation constructed inductively using the above clauses. A *regular model* over Π is a model built over a regular frame (that is, regular models are regular frames together with a valuation). When working with PDL over the programs in Π, we will be interested in regular models for Π, for these are the models that capture the intended interpretation. All very simple and natural — but by insisting that $R^{\pi_1^*}$ be interpreted by the reflexive transitive closure of R^{π_1}, we have given PDL genuinely *second-order* expressive power. A straightforward application of the Compactness Theorem shows that first-order logic cannot define the transitive closures of arbitrary binary relations, so with this definition we've moved beyond the confines of first-order logic. Unsurprisingly, compactness fails in PDL. To see this, consider the following infinite set of formulas:

$$\{\langle \pi^* \rangle p, \neg p, [\pi]\neg p, [\pi][\pi]\neg p, [\pi][\pi][\pi]\neg p, \ldots\}.$$

It is clear that every finite subset of this set has a regular model: we simply make p true at a state reachable by taking $n+1$ (non-reflexive) π-steps out from the current state, where n is the maximal level of nesting of boxes. But the entire set cannot be satisfied at any state in any regular model.

So we have genuine second-order expressivity at our disposal. What can we do with it? Well, for a start, at the level of models, we can express some familiar algorithmic constructs:

$(p? \,; a) \cup (\neg p? \,; b)$	`if` p `then` a `else` b.
$a; (\neg p?; a)^*; p?$	`repeat` a `until` p.
$(p?; a)^*; \neg p?$	`while` p `do` a.

Note the crucial role played by * in capturing the effect of the two loop constructors.

Moreover, the second-order expressivity built in at the level of models spills over into the level of frames. Here's a nice illustration. Via the concept of validity, PDL itself is strong enough to

define the class of regular frames (something which cannot be done in a first-order language). Now, it is not hard to give conditions that capture choice and composition. The formula

$$\langle \pi_1 \cup \pi_2 \rangle p \leftrightarrow \langle \pi_1 \rangle p \vee \langle \pi_2 \rangle p$$

is valid on precisely those frames satisfying $R^{\pi_1 \cup \pi_2} = R^{\pi_1} \cup R^{\pi_2}$, and

$$\langle \pi_1 ; \pi_2 \rangle p \leftrightarrow \langle \pi_1 \rangle \langle \pi_2 \rangle p$$

is valid on precisely those frames satisfying $R^{\pi_1 ; \pi_2} = R^{\pi_1} \circ R^{\pi_2}$.

But these are first-order conditions. What about iteration? We demanded that the relation R^{π^*} used for the program π^* be the reflexive transitive closure of the relation R^{π} used for π. This constraint cannot be expressed in first-order logic; how can we impose it via PDL validity?

As follows. First we demand that

$$\langle \pi^* \rangle \varphi \leftrightarrow \varphi \vee \langle \pi \, ; \pi^* \rangle \varphi$$

be valid. This says that a state satisfying φ can be reached by executing π a finite number of times if and only if φ is satisfied in the current state, or we can execute π once and then find a state satisfying φ after finitely many more iterations of π. Second, we demand that

$$[\pi^*](\varphi \rightarrow [\pi]\varphi) \rightarrow (\varphi \rightarrow [\pi^*]\varphi)$$

be valid too. This is called *Segerberg's axiom*. Work through what it says: as you will see, in essence it is an induction schema. A frame validates all instances of the four schemas just introduced if and only if it is a regular frame.

Summing up, at both the level of models and frames, PDL has a great deal of expressive power. Hence the following result is all the more surprising:

THEOREM 44. *PDL has the finite model property and is decidable. Its satisfiability problem is EXPTIME-complete.*

Proof. The finite model property, decidability, and EXPTIME-hardness results for PDL were proved in Fisher and Ladner [47, 48]. The existence of an EXPTIME algorithm for PDL satisfiability was proved in Pratt [103]. ❑

But we are only half-way through our story. With the modal μ-calculus we will climb even higher in second-order expressivity hierarchy, and we will do so without leaving EXPTIME.

6.7 *The modal μ-calculus*

The modal μ-calculus is the basic modal language extended with a mechanism for forming least (and greatest) fixed-points. It is highly expressive (as we shall see, it is stronger than PDL) and computationally well behaved. Moreover it has a beautiful bisimulation-based characterisation. All in all, it is one of the most significant languages on the modal landscape. It was introduced in its present form by Dexter Kozen [80].

The idea underlying the modal μ-calculus is to view modal formulas as *set-theoretic operators*, and to add mechanisms for specifying their fixed-points. Now, a set-theoretic operator on a set W is simply a function $F : 2^W \mapsto 2^W$. But how can we view modal formulas as set-theoretic operators? Consider a formula φ containing some proposition symbol (say p). In any

model, φ will be satisfied at some set of points. If we systematically vary the set of points that the valuation assigns to p, the set of points where φ is satisfied will typically vary too. So we can view φ as inducing an operator over the points of some model, namely the operator that takes as argument the subset of W that is assigned to p, and returns the set of points where φ is satisfied with respect to this assignment.

Let's make this precise. We will work in a language with a collection of diamonds $\langle\pi\rangle$, so models have the form $\mathfrak{M} = (W, \{R^\pi\}_{\pi\in\mathrm{MOD}}, V)$. For any proposition symbol p, $V(p)$ is the set of points in $\mathfrak{M}$ where p is satisfied. Let's extend V to a function that returns, for arbitrary formulas φ, the set of points in $\mathfrak{M}$ that satisfy φ (we won't invent a new name for this extended valuation, we'll simply call it V). The required definition is a simple reformulation of the satisfaction definition for the basic modal language:

$$\begin{array}{rcl} V(p) & = & V(p) \text{ for all proposition symbols } p \\ V(\neg\varphi) & = & W \backslash V(\varphi) \\ V(\varphi\wedge\psi) & = & V(\varphi)\cap V(\psi) \\ V(\langle\pi\rangle\varphi) & = & \{w \mid \text{for some } v\in W,\ R^\pi wv \text{ and } v\in V(\varphi)\}. \end{array}$$

Furthermore, for any proposition symbol p and any $U \subseteq W$ we shall write $V_{[p\leftarrow U]}$ for the (extended) valuation that differs from the (extended) valuation V, if at all, only in that it assigns U to p. That is, $V_{[p\leftarrow U]}(p) = U$, and for any $q \neq p$, $V_{[p\leftarrow U]}(q) = V(q)$. Then the operator induced by a formula φ (relative to a proposition symbol p) is the function that maps any $U \subseteq W$ to $V_{[p\leftarrow U]}(\varphi)$.

Now to bring fixed-points into the picture. A subset X of W is a fixed-point of a set-theoretic operator F on W if $F(X) = X$. This is clearly a special property: which set-theoretic operators have fixed-points, and how do we calculate them? The Knaster-Tarski Theorem (see Knaster [79] and Tarski [123]) gives important answers. Firstly, this theorem tells us that fixed-points exist when we work with *monotone* set-theoretic operators (an operator F is monotone if $X \subseteq Y$ implies that $F(X) \subseteq F(Y)$). Secondly, this theorem tells us that if F is a monotone operator on a set W, then F has a least fixed-point μF, which is equal to

$$\bigcap\{U\subseteq W \mid F(U)\subseteq U\},$$

and also a greatest fixed-point νF, which is equal to

$$\bigcup\{U\subseteq W \mid U\subseteq F(U)\}.$$

That is, both μF and νF are solutions to the equation $F(X) = X$, and furthermore, for any other solution Z, we have that $\mu F \subseteq Z \subseteq \nu F$. The least and greatest fixed-points given by the Knaster-Tarski Theorem are the fixed-points the modal μ-calculus works with.

But how can we specify these fixed-points using modal formulas? By enriching the syntax with an operator μ that binds occurrences of proposition symbols. That is, we shall write expressions like $\mu p.\varphi$, in which all free occurrence of the proposition symbol p in φ are bound by μ. The intended interpretation of $\mu p.\varphi$ is that it denotes the subset of W that is the least fixed-point of the set-theoretic operator induced by φ with respect to p. Fine — but how do we know that this fixed-point exists? If φ is arbitrary, we don't. However if all free occurrences of p in φ occur positively (that is, if they all occur under the scope of an even number of negations) then a simple inductive argument shows that the set-theoretic operator induced by φ is monotone, and hence (by the Knaster-Tarski Theorem) has least (and greatest) fixed-points. Accordingly we impose

the syntactic restriction that the μ operator can only be used to bind a proposition symbol when all free occurrences of the variable occur positively. With this restriction in mind we define:

$$V(\mu p.\varphi) \;=\; \bigcap\{U \subseteq W \mid V_{[p \leftarrow U]}(\varphi) \subseteq U\}.$$

That is, the set assigned to $\mu p.\varphi$ is the least fixed-point of the operator induced by φ.

What can we say with the modal μ-calculus? Consider the expression

$$\mu p.(\varphi \vee \langle\pi\rangle p).$$

Read this as defining "the least property (subset) p such that either φ is in p or $\langle\pi\rangle p$ is in p". What is this set? A little experiment will convince you that it must be

$$\{w \in W \mid \mathfrak{M}, w \models \varphi \text{ or there is a finite } R^{\pi}\text{-sequence from } w \text{ to } v \text{ such that } \mathfrak{M}, v \models \varphi\}.$$

(The reader should check that this set really is the one given to us by the Knaster-Tarski Theorem.) Note that this is exactly the set of points that make the PDL formula $\langle\pi^*\rangle\varphi$ true.

How do we specify greatest fixed-points? With the help of the ν operator. This is defined as follows:

$$\nu p.\varphi \;=_{\mathit{def}}\; \neg\mu p.\neg\varphi(\neg p/p),$$

where $\varphi(\neg p/p)$ is the result of replacing occurrences of p by $\neg p$ is φ. This expression is well-formed: if φ is a formula that we could legitimately apply the μ operator to (that is, if all occurrences of p occur under the scope of an even number of negations), then so is $\neg\varphi(\neg p/p)$. The reader should check that this operator picks out the following set:

$$V(\nu p.\varphi) \;=\; \bigcup\{U \subseteq W \mid U \subseteq V_{[p \leftarrow U]}(\varphi)\}.$$

That is (in accordance with the Knaster-Tarski Theorem) it picks out the greatest fixed-point of the operator induced by φ. As a further exercise, the reader should check that

$$\nu p.(\varphi \wedge [\pi]p)$$

denotes the following set:

$$\{w \in W \mid \mathfrak{M}, w \models \varphi \text{ and at every } v \text{ reachable from } w \text{ by a finite } R^{\pi}\text{-sequence, } \mathfrak{M}, v \models \varphi\}.$$

Note that this is exactly the set of points w that make the PDL formula $[\pi^*]\varphi$ true.

In view of these examples, it should not come as a surprise that PDL can be translated into the modal μ-calculus. Here are the key clauses:

$$\begin{array}{rcl}
(\langle\pi_1 \cup \pi_2\rangle\varphi)^{mu} & = & \langle\pi_1\rangle(\varphi)^{mu} \vee \langle\pi_2\rangle(\varphi)^{mu} \\
(\langle\pi_1 ; \pi_2\rangle\varphi)^{mu} & = & \langle\pi_1\rangle\langle\pi_2\rangle(\varphi)^{mu} \\
(\langle\pi^*\rangle\varphi)^{mu} & = & \mu p.((\varphi)^{mu} \vee (\langle\pi\rangle p)^{mu}), \text{ where } p \text{ does not occur in } \varphi.
\end{array}$$

In fact the modal μ-calculus is strictly more expressive than PDL. The simplest example of a construct that PDL cannot model but that the modal μ-calculus can is the *repeat* operator. The expression *repeat(π)* is true at a state w if and only if there is an infinite sequence of R^{π} transitions leading from w. Proving that this is not expressible in PDL is tricky, but it can be expressed in the modal μ-calculus: the formula $\nu p.\langle\pi\rangle p$ does so. Moreover, the temporal logics

standardly used in computer science, such as LTL, CTL, and CTL*, can also be embedded in the modal μ-calculus. For remarks and references on this topic, see Chapter 12 of this handbook.

All in all, the modal μ-calculus is a highly expressive language. In spite of this, it is extremely well behaved, both computationally and in other respects. For a start we have that:

THEOREM 45. *The modal μ-calculus has the finite model property and is decidable. Its satisfiability problem is EXPTIME-complete.*

Proof. The original decidability proof was given in Kozen and Parikh [81]. The finite model property was first established in Street and Emerson [119]. The complexity result is from Emerson and Jutla [36]. ❑

Furthermore, experience shows that the modal μ-calculus is also well behaved when it comes to model checking — indeed it is widely believed that its model checking task can be performed in polynomial time. However, at the time of writing, this conjecture has resisted all attempts to prove it.

Moreover, the modal μ-calculus has an elegant semantic characterisation. Suppose we add the following clause to the standard translation for basic modal logic:

$$\mathrm{ST}_x(\mu p.\varphi) = \forall P(\forall y((\mathrm{ST}_x(\varphi) \to Py) \to Py)).$$

This clearly captures the intended semantics of μ. But note that by adding this clause we are viewing the standard translation as taking us to monadic second-order logic, for here we bind the unary predicate symbol P. This language is already familiar to us: it's the frame correspondence language introduced in Section 5, but here we're using it to express a correspondence at the level of *models*. Thus (even at the level of models) the modal μ-calculus is a fragment of monadic second-order logic. But which fragment? This one:

THEOREM 46 (Modal μ-Calculus Characterisation Theorem). *The modal μ-calculus is the bisimulation invariant fragment of monadic second-order logic.*

Proof. See Janin and Walukiewicz [73]. ❑

For more on the modal μ-calculus, see Chapter 12 of this handbook. As well as giving a detailed technical overview, the chapter also gives an informal introduction to thinking in terms of fixed-points, which is often a stumbling block when the modal μ-calculus is encountered for the first time.

6.8 Combined logics

We now turn to what is (at first glance) one of the simplest methods of obtaining a richer modal language: combine two pre-existing ones. But for all its apparent simplicity, this method of enrichment swiftly leads to difficult territory.

Many applications lead naturally to the idea of combined logics. A good example is planning. Planning involves a collection of agents who must reason about what they are going to do given that they know the effects of actions, and where getting more information may be important for solving the problem at hand. Hence Robert Moore [98] proposed a combined language for this task. His language offered both epistemic and action modalities, making it possible to say things like

$K_i[a]\varphi$ "agent i knows that doing a has the effect φ"

and

$$[a]K_i\varphi \quad \text{“doing } a \text{ makes agent } i \text{ know that } \varphi\text{”.}$$

Actually, Moore also considered combinations of PDL with epistemic operators, as plans are usually complex actions with program structure.

The fun starts when we ask how the two logics live together. For example, should they simply live side by side, the simple fusion of the two component logics? Or are there interactions between them? Obviously this depends on what we are modeling. For example, should $K_i[a]\varphi$ imply $[a]K_i\varphi$? In general, no. After all, I may know that after drinking I am boring, but unfortunately after drinking I no longer know that I am boring (that is, drinking is not an epistemically transparent action). Nor need the converse implication hold for actions that deliver genuinely new information. After consulting my account manager, I know I am broke, but I do not know now that after the consultation I am broke.

If our application does not require the modeling of such interactions, then we are dealing with the simplest possible combination of two decidable modal logics, and the result is again decidable. But for some applications we might want to enforce these interactions. Let R_a be the accessibility relation for action a, and let $\sim_i$ be the epistemic relation for agent i. The following frame correspondences tell us what these interactions give rise to:

$$\begin{array}{ll} \mathfrak{F} \models K_i[a]p \to [a]K_ip & \text{iff } \forall xyz((R_axy \wedge y \sim_i z) \to \exists u(x \sim_i u \wedge R_auz)) \\ \mathfrak{F} \models [a]K_ip \to K_i[a]p & \text{iff } \forall xyz((x \sim_i y \wedge R_ayz) \to \exists u(R_axu \wedge u \sim_i z)). \end{array}$$

The first principle says that new uncertainty links between the results of an action are inherited from existing ones; this is a version of the game-theoretic principle of *perfect recall*. The other direction is called *no learning*. These are powerful interaction principles. Indeed, they impose a grid-like interaction between the relations interpreting the modalities, hence the possibility arises of showing undecidability by encoding the tiling problem. A good source of information on this topic is Halpern and Vardi [64]. Among other things they show that the combined modal epistemic logic of agents with perfect recall, though still decidable, is highly complex, and that if a common knowledge operator (that is, using PDL notation, a box of the form $[(\sim_1 \cup \cdots \cup \sim_n)^*]$) is added, the problem becomes undecidable. This is a natural example of the bad computational behaviour that combinations of relatively simple decidable modal logics can give rise to. Moreover the air of mystery (“How can a description of well behaved agents get so complex?”) quickly gets dispelled once we realise that the behaviour of special agents may have a rich mathematical structure that makes their logic tough.

In recent years there has been intensive theoretical work on combinations of modal logic. The goal has been to provide general *transfer results*: given two (or more) modal logics, and a method of combining them, when do properties such as decidability, finite model property, and finite axiomatisability transfer from the component logics to the combined logic? The simplest way of combining two modal logics is to take their *fusion*. Given two modal logics L_1 and L_2 (in languages with disjoint sets of modal operators) then their fusion $L_1 \otimes L_2$ is the smallest logic L in their joint language that contains them both. Fusions of modal logic have been investigated in detail (key papers include Kracht and Wolter [82], Fine and Schurz [46], and Wolter [144]), and have some pleasant transfer properties. For example, to axiomatise the fusion logic L, it suffices to take the axioms for each of the components (that is, no interaction axioms involving modalities from both language are required). Moreover, both the finite model property and decidability transfer from the component logics to the fusion.

But this good behaviour reflects the fact that fusion is a combination method designed to minimise the interaction between the component modalities. What of combination methods

which allow strong interaction between the modalities? The best studied combination technique here is the formation of *products* of modal logics. Given two frames $\mathfrak{F}_1 = (W_1, R_1)$ and $\mathfrak{F}_2 = (W_2, R_2)$, their product $\mathfrak{F}_1 \times \mathfrak{F}_2$ is the frame $(W_1 \times W_2, R_h, R_v)$. Here R_h is the binary relation on $W_1 \times W_2$ defined by $(u_1, v_1)R_h(u_2, v_2)$ iff $u_1 R_1 u_2$ and $v_1 = v_2$; and R_v is the relation defined by $(u_1, v_1)R_v(u_2, v_2)$ iff $v_1 R_2 v_2$ and $u_1 = u_2$. The idea of taking products of modal logics is an old one (dating back to at least Segerberg [114]) and is a widely used combination method in many applications of modal logic. But the product construction creates frames which allow for very strong interactions between the modalities, and there are far fewer transfer results for this method of combination; indeed, there are many negative results showing transfer of decidability failures.

Work on combination of logics, from both applied and theoretical perspectives, is one of the liveliest areas of research in contemporary modal logic. For a detailed survey of fusions, products, and methods of combinations between these extremes, see Chapter 15 of this handbook.

6.9 First-order modal logic

We turn now to what is arguably one of the least well behaved modal languages ever proposed: first-order modal logic. However, in one of those twists that make intellectual history so fascinating, first-order modal logic has come to be accepted (at least in philosophical quarters) as the most important modal logic of all. For many philosophers, modal logic *is* first-order modal logic.

This is not to say that first-order modal logic is philosophically uncontroversial. Indeed, as is discussed in Chapter 21 of this handbook, one of the liveliest debates in 20th century analytic philosophy was ignited when Quine [106] questioned the coherence of the enterprise. But two advances led to its acceptance. The first was the development of the relational semantics of first-order modal logic (Kripke [83, 85] are key papers here) and the second was the publication of “Naming and Necessity” (Kripke [86]) which presented what is probably the most widely accepted philosophical interpretation of the technical machinery. While these developments did not dispel all the controversy, nowadays first-order modal logic together with (some form of) relational semantics, is generally regarded as a well understood (perhaps even boringly familiar) tool of philosophical analysis.

Viewed from a mathematical perspective, however, things look rather different. Had first-order modal logic never existed, a logician who proposed its (now standard) syntax and relational semantics might have been regarded as audacious, perhaps downright careless. Why? Because, in essence, first-order modal logic is a combined logic. As we have just seen, combining two modal logics while retaining interesting properties is no easy matter. So it should not come as too much of a surprise that combining propositional modal logic with first-order logic is unlikely to be plain sailing. In what follows we shall sketch the standard syntax and semantics, and mention some of its problematic features.

First the syntax (we omit some of the clauses for the booleans):

$$\varphi \quad ::= \quad P(x_1, \ldots, x_n) \mid x = y \mid \neg\varphi \mid \varphi \to \psi \mid \Diamond\varphi \mid \Box\varphi \mid \exists x\varphi \mid \forall x\varphi.$$

Here P is an n-place predicate symbol and the x_i are individual variables. So (given the clauses for the quantifiers and booleans) it is clear that we have a full first-order language at our disposal, and hence (because of the presence of the modalities) we can now search for first-order information at accessible states in the familiar way. But we can do more. The clauses for the quantifiers hide a subtlety: if a formula φ contains free first-order variables within the scope of a

modality, then formulas of the form $\forall x\varphi$ and $\exists x\varphi$ bind variables within the scope of the modality. This possibility is what led to Quine's philosophical objections ("no binding into intensional contexts"). And from a technical perspective it means we are combining two very different styles of logic in a way that allows a strong form of interaction.

The standard semantics for first-order modal logic comes in a number of variant forms. One basic choice concerns the domain of quantification: should the quantifiers range over some fixed domain of quantification (the *constant domain* semantics), or should each point be associated with its own domain (the varying domain semantics)? Here we shall present the varying domain semantics; for a discussion of the constant domain approach, and of equivalences between the constant domain, varying domain, and other approaches, see Chapter 9 of this handbook, or Fitting and Mendelsohn [49].

DEFINITION 47. A varying domain model is a tuple $(W, R, D, \{\delta_w\}_{w\in W}, \{V_w\}_{w\in W})$. Here W is a non-empty set; R is a binary relation on W; D (the domain of quantification) is a non-empty set; for all $w \in W$, $\delta_w \subseteq D$; and for all $w \in W$, V_w is a function that assigns to each n-place predicate symbol a subset of D^n.

That is, we have the familiar modal machinery from the propositional case (note that (W, R) is just a frame, and the V_w are essentially our familiar valuations upgraded to interpret first-order n-place predicate symbols P rather than proposition symbols p) augmented by a specification (the δ_w) of the individuals the quantifiers at each state w range over. We interpret first-order modal logic by taking such a model, together with an assignment of values to variables (that is, a function g that maps the individual variables to elements of D), and using the following satisfaction definition:

$$
\begin{array}{rll}
\mathfrak{M}, g, w \models P(x_1, \ldots, x_n) & \text{iff} & (g(x_1), \ldots g(x_n)) \in V_w(P), \\
\mathfrak{M}, g, w \models x = y & \text{iff} & g(x) = g(y), \\
\mathfrak{M}, g, w \models \neg\varphi & \text{iff} & \text{not } \mathfrak{M}, g, w \models \varphi, \\
\mathfrak{M}, g, w \models \varphi \rightarrow \psi & \text{iff} & \mathfrak{M}, g, w \not\models \varphi \text{ or } \mathfrak{M}, g, w \models \psi, \\
\mathfrak{M}, g, w \models \Diamond\varphi & \text{iff} & \text{for some } v \in W \text{ such that } Rwv \text{ we have } \mathfrak{M}, g, v \models \varphi, \\
\mathfrak{M}, g, w \models \Box\varphi & \text{iff} & \text{for all } v \in W \text{ such that } Rwv \text{ we have } \mathfrak{M}, g, v \models \varphi, \\
\mathfrak{M}, g, w \models \exists x\varphi & \text{iff} & \text{for some } g' \sim_x g \text{ where } g'(x) \in \delta_w \text{ we have } \mathfrak{M}, g', v \models \varphi, \\
\mathfrak{M}, g, w \models \forall x\varphi & \text{iff} & \text{for all } g' \sim_x g \text{ such that } g'(x) \in \delta_w \text{ we have } \mathfrak{M}, g', v \models \varphi.
\end{array}
$$

(Here $g' \sim_x g$ means that the assignments g and g' are identical save possibly in the value they assign to the variable x.)

This language is capable of expressing some important distinctions. Consider, for example, the formulas $\forall x\Box\varphi$ and $\Box\forall x\varphi$. The first asserts, of each existing entity, that it has the property φ at all accessible states. The second asserts that, at each accessible state, each entity that exists at that particular state has property φ. Should either of these formulas imply the other? That is, should we accept as valid either of the following two principles?

$$
\begin{array}{ll}
\forall x\Box\varphi \rightarrow \Box\forall x\varphi & \text{Barcan formula} \\
\Box\forall x\varphi \rightarrow \forall x\Box\varphi & \text{Converse Barcan formula}
\end{array}
$$

Instead of trying to answer such tricky philosophical questions (which bear on the *de dicto*/*de re* distinction, discussed in Chapter 9 of this handbook) let us consider what they say in the light of

the relational interpretation just given. It is not difficult to see that the Barcan formula is valid in a varying domain model iff that model has *decreasing domains*, that is, if for all $w, v \in W$, Rwv implies $\delta_v \subseteq \delta_w$. And the Converse Barcan formula is valid on precisely *increasing domain* models, that is, models with the property that Rwv implies $\delta_w \subseteq \delta_v$. So to insist on the validity of both principles is to force an even stronger interaction between the quantifiers and modalities: it takes us to a locally constant domain semantics in which Rwv implies $\delta_w = \delta_v$. This is a good example of the clarity that relational semantics can bring to difficult conceptual issues, and shows why first-order modal logic can be useful in philosophical logic and natural language semantics.

So what's the problem? Simply this: for all its analytical utility, first-order modal logic under its standard semantics is not well behaved mathematically. Early signs of trouble appeared in Fine [45], which showed that interpolation and the Beth property fail for first-order **S5** under the varying domain semantics, and for any first-order modal logic between **K** and **S5** under the constant domain semantics. As **S5** is both philosophically central (it is widely considered to embody the logic of "necessarily" and "possibly") and semantically straightforward (it is the logic of frames in which R is an equivalence relation) these are strong negative results indeed. Worse was to come. It turns out that it is possible to take a propositional modal logic that is complete with respect to some class of frames, axiomatically extend it in the manner naturally suggested by the standard semantics, and yet to wind up with an incomplete first-order modal logic (see Ghilardi [56], Shehtman and Skvortsov [117], Corsi and Ghilardi [26], Cresswell [27]).

Now, the issue here is not so much the incompleteness in itself (as we have already discussed, even in the propositional modal logic, frame incompleteness results are the norm) rather it is the *loss* of completeness in the transition from the propositional case to the first-order case that is worrying. To use the terminology introduced when we discussed combinations of logics: the standard relational semantics for first-order logic is a method of combination for which transfer of completeness fails.

Such results have led to renewed technical interest in first-order modal logic. The semantics of first-order modal logic has come under intense scrutiny, and a number of alternative semantics have been proposed which enable completeness results to be transferred. Some of this work has been model-theoretic (see, in particular, van Benthem's [132] use of functional frames) but most of it has been highly abstract, employing the language of category theory; for a detailed account of such work, see Chapter 9 of this handbook. More recently, the hybrid logic community has pointed out that upgrading the underlying propositional modal language to a hybrid language is another way to repair the situation: interpolation is regained (see Areces, Blackburn and Marx [7]), indeed, regained constructively (see Blackburn and Marx [15]) and general positive results on transfer of completeness can be proved (see Blackburn and Marx [14]). All in all, first-order modal logic is one of the most intriguing areas of modal logic: the most venerable system of all poses some of the most interesting questions about what it is to be modal.

6.10 General perspectives

Moving to richer languages better fitted for particular applications is a standard feature of current research. It is true that in some quarters sticking to the poorest modal base language of the founding fathers (despite its evident handicaps in expressive power and mathematical convenience) is still something of a religion. But the idea of designing extensions is not some new-fangled notion; its roots stretch back to the work of von Wright [145] and Prior [104, 105], and the idea was central to the work of the Sofia School (see, for example, Passy and Tinchev [101] for insightful comments on what modal logic is and why one might want to enrich it). Still, pointing to a

noble heritage is not enough. We need to address a tricky question: what makes these languages *modal*? Being precise here is difficult. As we have seen, there is a wide range of extensions. Moreover, each application imposes its own concerns and peculiarities. Nevertheless, there is a guiding idea that lies behind most examples of this form of language design: obtaining a reasonable balance between expressive power and computational complexity. So the question we should focus on is: what makes such natural balances arise?

As we have seen, many richer modal languages are fragments of the full language of first-order logic over some appropriate similarity type of relations and properties. We can see this by translation, just as we did with the basic modal language (we saw that the complex truth conditions for the Until and Since are definable by first-order formulas, and the same is true for the conditional connective, the universal modality, and the apparatus of hybrid logic). Now, there have been various attempts to find general patterns explaining which parts of first-order logic are involved in modal languages. Gabbay [51] observed that modal languages tend to translate into so-called *finite variable fragments* of first-order logics, that is, fragments using only some finite number of variables, fixed or bound. For example, we have seen that the basic modal language can make do with only two variables, and temporal logic with Until and Since, and conditional logic, only require three. Finite variable fragments have some pleasant computational behaviour; for example, their model checking complexity is in PTIME (see Vardi [141]) as opposed to PSPACE for the full first-order language. On the other hand, as we have already mentioned, satisfiability is already undecidable for first-order fragments with three variables, so the real reason for the low complexity of modal languages lies elsewhere. A different type of analysis for the latter phenomenon was given in the paper "Why is modal logic so robustly decidable?" (Vardi [142]). This emphasises the semantic adequacy of the tree-like models obtainable via bisimulation unraveling of arbitrary graph models. This type of explanation is important as it transcends first-order logic; on the other hand it does not provide much in the way of concrete syntactic insight. For the latter, the current best explanation is the one provided by the guarded fragment and its relatives (which are, arguably, the strongest known modal languages).

As we saw, guarded fragments locate the essence of modal logic in the *restriction* on the quantification performed by the modalities. One attractive property of this analysis is its logical resilience: it turns out that it extends beyond the setting of first-order enrichments to second-order enrichment too, something that was not forseen when the guarded fragment was first isolated. A striking example is the result in Grädel and Walukiewicz [63] that the extension of the guarded fragment with the fixed-point operators μ and ν remains decidable. By way of contrast, validity for full first-order logic extended with these operators is non-axiomatisable, indeed, non-arithmetical. This observation shows that the modal philosophy embodied in the idea of guarded fragments is not restricted to first-order extensions: often modal fragments can bear the weight of additional higher-order apparatus (such as fixed-point operators) which would send the full first-order correspondence languages into a tailspin complexity wise. Our discussion of PDL and the modal μ-calculus has shown that this is the case for the basic modal language. Grädel and Walukiewicz's result for the guarded fragment shows that this type of behaviour persists higher up: guarded quantification can support higher-order constructions too.

Perhaps guarding can be a fruitful strategy in even more exotic modal settings? One setting worth exploring is *infinitary* modal logic. This logic (which was used extensively in Barwise and Moss [10] and Baltag [8] for investigating non-well founded set theory; see Chapter 16 of this handbook) provides a perfect match with bisimulation: two pointed models are bisimilar if and only if they satisfy the same formulas in a modal language that allows arbitrary infinite conjunctions and disjunctions. Moreover a modal characterisation theorem holds. Now, decidability is a

non-issue in this setting, but what about existential semantic properties such as interpolation and Beth Definability? It is known that interpolation holds for infinitary modal logic (see Barwise and van Benthem [11]), but can such results be lifted to infinitary guarded fragments? Another setting worth exploring in this way is *second-order propositional modal logic*, in which we can quantify over proposition symbols (see Fine [42] for some early results, ten Cate [124] for a more recent discussion, and Chapter 10 of this handbook for a brief overview). The equation "modality = guarding" should be simultaneously regarded as a hypothesis to be tested in richer settings, and as a useful heuristic for isolating further logics worth calling modal.

Not that we should put all our eggs in one basket. Perhaps the notion of modality is too diffuse for any single approach to exhaust, and in any case it is worth looking for alternatives. Another approach is to apply ideas from abstract model theory (see Barwise and Feferman [9]). This was first done in de Rijke [30], who proved a modal analog of Lindström's [91] celebrated characterisation of first-order logic. The original form of Lindström's theorem says that an abstract logic $\mathcal{L}$ extending first-order logic coincides with first-order logic iff it has the compactness and Löwenheim-Skolem properties. Another way of stating the theorem is that an abstract logic $\mathcal{L}$ extending first-order logic coincides with first-order logic iff it has the compactness and Karp properties. (The Karp property is that all formulas are invariant for potential isomorphism, where a potential isomorphism is a non-empty family of finite partial isomorphisms closed under the usual back and forth extension properties; recall our discussion of partial isomorphisms in Section 3.3). We shall discuss a (slightly reformulated) version of de Rijkes's result and a more recent characterisation due to van Benthem.

What is an abstract modal logic? Here's the conception that underlies our reformulation of de Rijke's result. We give it in terms of pointed models $(\mathfrak{M}, w)$, that is, a model together with a point of evaluation.

DEFINITION 48 (Very abstract modal logics). Let $\mathcal{L}$ be a set of formulas, and $\models_{\mathcal{L}}$ its satisfaction relation, that is, a relation between pointed models and $\mathcal{L}$-formulas. A very abstract modal logic is a pair $(\mathcal{L}, \models_{\mathcal{L}})$ with the following properties:

1. Occurrence property. For each φ in $\mathcal{L}$ there is an associated finite language $\mathcal{L}(\lambda_\varphi)$. The relation $(\mathfrak{M}, w) \models_{\mathcal{L}} \varphi$ is a relation between $\mathcal{L}$-formulas φ and models $(\mathfrak{M}, w)$ for languages $\mathcal{L}$ containing $\mathcal{L}(\lambda_\varphi)$. That is, if φ is in $\mathcal{L}$, and $\mathfrak{M}$ is an $\mathcal{L}$-model, then $(\mathfrak{M}, w) \models_{\mathcal{L}} \varphi$ is either true or false if $\mathcal{L}(\lambda_\varphi) \subseteq \mathcal{L}$, and undefined otherwise.

2. Expansion property. The relation $(\mathfrak{M}, w) \models_{\mathcal{L}} \varphi$ depends only on the restriction of $\mathfrak{M}$ to $\mathcal{L}(\lambda_\varphi)$. That is, if $(\mathfrak{M}, w) \models_{\mathcal{L}} \varphi$ and $(\mathfrak{N}, w)$ is an expansion of $(\mathfrak{M}, w)$ to a larger language, then $(\mathfrak{N}, v) \models_{\mathcal{L}} \varphi$.

A very abstract modal logic $(\mathcal{L}, \models_{\mathcal{L}})$ extends basic modal logic if for every basic modal formula there exists an equivalent $\mathcal{L}$-formula (that is, if for each basic modal formula φ there exists an $\mathcal{L}$-formula ψ such that for any model $(\mathfrak{M}, w)$ we have $(\mathfrak{M}, w) \models \varphi$ iff $(\mathfrak{M}, w) \models_{\mathcal{L}} \psi$).

De Rijke's characterisation centres on the familiar bisimulation invariance property and the *finite depth property*. A very abstract modal logic $\mathcal{L}$ has the *finite depth property* iff for any $\mathcal{L}$-formula φ there is some natural number k such that for all models $\mathfrak{M}$,

$$\mathfrak{M}, w \models \varphi \text{ iff } \mathfrak{M}|k, w \models \varphi,$$

where $\mathfrak{M}|k$ is the model $\mathfrak{M}$ restricted to just those points that can be reached from w in k or fewer R-steps. De Rijke builds invariance for bisimulation into the notion of abstract modal

logic, so his statement of his Lindström-style result has the form: any abstract modal logic with the finite depth property that extends the basic modal language is the basic modal language. Reformulating his result in terms of very abstract modal logics, thereby making the bisimulation invariance condition explicit, results in:

THEOREM 49. *Suppose $\mathcal{L}$ is a very abstract modal logic extending the basic modal language. Then $\mathcal{L}$ coincides with the basic modal language iff $\mathcal{L}$ has the finite depth and invariance for bisimulation properties.*

Proof. See de Rijke [30, 31]. For a textbook-level exposition of the proof, see Theorem 7.60 of Blackburn, de Rijke and Venema [13]. ❑

This is an informative result. Nonetheless, the finite depth property seems somewhat engineered to capture the basic modal language, and it is natural to look for generalisations. However, because of the expressive limitations of modal languages, this is not straightforward. The proof of the Lindström Theorem for first-order logic typically proceeds by contradiction: to show that an abstract first-order formula has a first-order equivalent, one typically builds a model where φ is true in one part, $\neg\varphi$ in another, and uses the expressive power of first-order logic to link the two parts of the model by a chain of partial isomorphisms, thereby reaping the contradiction. This style of argument does not lift easily to modal languages: the basic modal language is too impoverished to encode the chains of bisimulations linking the two parts of the model that would be required to mimic this proof technique directly. However, as van Benthem [139] observed, there is a way around this. The key idea is to strengthen the definition of a very abstract modal language by demanding it fulfils the *relativisation* condition:

DEFINITION 50 (Abstract modal logics). An abstract modal logic $\mathcal{L}$ is a very abstract modal logic that has the *relativisation property*: for any $\mathcal{L}$-formula φ and proposition symbol p not occurring in φ, there is a formula $\mathit{Rel}(\varphi, p)$ which is true at a model $(\mathfrak{M}, w)$ iff φ is true at $(\mathfrak{M}|p, w)$, which is the submodel of $\mathfrak{M}$ consisting of just those points that satisfy p.

Relativisation is a natural property (most logics satisfy it) but the key point is to observe is how it is used in the proof of the following theorem: in essence, it provides a model-theoretic tool which enables us to give an alternative proof without resorting to explicit codings of bisimulations. This leads to van Benthem's version of the Lindström Theorem for modal logic:

THEOREM 51. *Suppose $\mathcal{L}$ is an abstract modal logic extending the basic modal language. Then $\mathcal{L}$ coincides with the basic modal language iff $\mathcal{L}$ satisfies compactness and invariance for bisimulation.*

Proof. We know that the basic modal language satisfies compactness (Proposition 4) and invariance for bisimulation (Lemma 9) so the left to right direction is clear. For the reverse direction, assume that $\mathcal{L}$ has these properties. We claim that the following holds: *in a compact abstract modal logic $\mathcal{L}$ which is invariant for bisimulations, every formula has the finite depth property.* If we can show this, the result follows from Theorem 49.

We prove the claim as follows. Let φ be any formula in $\mathcal{L}$. Suppose for the sake of a contradiction that φ lacks the finite depth property. Then for any natural number k there exists a model $(\mathfrak{M}_k, w)$ and a cut-off version $(\mathfrak{M}_k|k, w)$ which disagree on the truth value of φ. Without loss of generality, assume that the following happens for arbitrarily large k: $(\mathfrak{M}_k|k, w) \models \varphi$, and $(\mathfrak{M}_k, w) \models \neg\varphi$ (here we use the fact that abstract modal logics are closed under negation). Now take a new proposition symbol p, and consider the following set Σ of $\mathcal{L}$-formulas:

$$\{\neg\varphi, \mathit{Rel}(\varphi, p)\} \cup \{\Box^n p \mid \text{for all natural numbers } n\}.$$

(By $\Box^n p$ we mean p prefixed by a sequence of n boxes.) Given our assumptions, this set is finitely satisfiable: we choose k sufficiently large, and make p true in the k reachable part of one of the above sequences of models. But then, by compactness for our abstract modal logic $\mathcal{L}$, there must be a model $(\mathfrak{N}, v)$ for the whole set Σ at once.

But this leads to a contradiction as follows. We focus on the generated submodel $(\mathfrak{N}_v, v)$ consisting of v and all points finitely reachable from it. Now, the identity relation is a bisimulation between any pointed model and its unique generated submodel. Hence, by the assumed invariance for bisimulation, formulas of $\mathcal{L}$ have the same truth value in any pointed model and its generated submodel. Now, given our definition of Σ, $\neg\varphi$ holds in $(\mathfrak{N}, v)$, and hence also in $(\mathfrak{N}_v, v)$. On the other hand, since $(\mathfrak{N}, v) \models Rel(\varphi, p)$, we have $(\mathfrak{N}|p, v) \models \varphi$. But by the truth of all the formulas of the form $\Box^n p$, p holds in the whole generated submodel $(\mathfrak{N}_v, v)$. Therefore we have that φ holds in (N_v, v). Contradiction. Hence the claim is established and the theorem follows. ❑

One surprising consequence of this result is that the Modal Characterisation Theorem (Theorem 13) follows from it; see van Benthem [139] for details.

It remains to be seen how widely applicable this technique is. For example, it is not straightforwardly applicable to languages with the universal modality, as these lack the finite depth property. However it *can* be lifted to the guarded fragment. As we mentioned in Section 6.5, there is a notion of guarded bisimulation. And using this notion, together with the relativisation technique leads to:

THEOREM 52. *Suppose $\mathcal{L}$ is an abstract modal logic extending the guarded fragment. Then $\mathcal{L}$ coincides with the guarded fragment iff $\mathcal{L}$ satisfies compactness and invariance for guarded bisimulation.*

Proof. See van Benthem [139]. ❑

7 ALTERNATIVE SEMANTICS

As we said at the start of this chapter, one of the most instructive ways of thinking about modal logic is to view it as a tool for talking about graphs. But to view modal logic exclusively through the lens of relational semantics would be a mistake; interesting alternatives exist, and in this section we introduce three of them: algebraic semantics, neighbourhood semantics, and topological semantics. As we shall see, each of these semantics has something new to offer. But we shall come across much that is familiar, for all three are linked in various ways with relational semantics.

7.1 Algebraic semantics

The basic idea of algebraic semantics is simple: view modal formulas as terms (or polynomials) and evaluate them in the appropriate type of algebra. So the key question is: what kinds of algebra are appropriate for modal logic? The answer is: *boolean algebras with operators*, or BAOs.

A *boolean algebra* is a triple $\mathfrak{A} = (A, +, \times, -, 1, 0)$ such that both $+$ (*join*) and $\times$ (*meet*) are commutative and associative binary operations, each of which distributes over the other. The unary operation $-$ (*complement*) must satisfy the equations $x + (-x) = 1$ and $x \times (-x) = 0$. The nullary operations (or *constants*) 1 and 0 must satisfy the equations $x \times 1 = x$ and $x + 0 = x$.

Even if you have never encountered boolean algebras before, a moment's reflection should make it clear that they are an algebraic mirror of propositional logic. To see this, read $+$ as $\vee$, $\times$ as $\wedge$, $-$ as $\neg$, 1 as $\top$, 0 as $\perp$, and $=$ as $\leftrightarrow$. So it only remains to provide algebraic structure that mirrors the diamonds. This motivates the following definition.

DEFINITION 53 (Boolean Algebras with Operators). A boolean algebra with operators, or BAO, is a pair $\mathfrak{B} = (\mathfrak{A}, m)$, where $\mathfrak{A}$ is a boolean algebra and m is a unary operator on $\mathfrak{A}$ that satisfies the equations $m(x + y) = m(x) + m(y)$, and $m(0) = 0$.

Note that the logical analogs of these two equations are $\Diamond(\varphi \vee \psi) \leftrightarrow (\Diamond\varphi \vee \Diamond\psi)$, and $\Diamond\perp \leftrightarrow \perp$, both of which are valid in relational semantics. Thus we now have an algebraic mirror for all components of the basic modal language.

We interpret the basic modal language in BAOs in the usual algebraic fashion. That is, given a BAO, we view the proposition symbols as variables ranging across the elements of the algebra, and interpret each logical operator by its corresponding algebraic operation. More precisely, let $\mathfrak{B}$ be a BAO, and V be a function mapping each proposition symbol to an element of $\mathfrak{B}$; we call such a function V an *algebraic valuation*. We extend V to a function that gives the result of evaluating arbitrary basic modal formulas in $\mathfrak{B}$ via the following recursive clauses:

$$\begin{array}{rcl} V(\varphi \vee \psi) & = & V(\varphi) + V(\psi) \\ V(\varphi \wedge \psi) & = & V(\varphi) \times V(\psi) \\ V(\neg\varphi) & = & -V(\varphi) \\ V(\Diamond\varphi) & = & mV(\varphi) \end{array}$$

It is now possible to prove the following algebraic completeness result:

THEOREM 54. *A basic modal formula belongs to the minimal modal logic **K** iff it evaluates to the value* 1 *in all modal algebras under all algebraic valuations.*

Proof. Straightforward. The key point is to use a technique standard in algebraic logic, namely to create the *Lindenbaum-Tarski Algebra* for **K**. The elements of the Lindenbaum-Tarski Algebra are equivalence classes of **K**-provably equivalent formulas; the operations are defined with the aid of the connectives. All and only the **K**-provable formulas evaluate to 1 in this algebra, and hence the result follows. For a detailed discussion, see Chapter 6 of this handbook. ❑

In fact, a far stronger result can be proved: *any* axiomatic extension of **K** (that is, *any* normal modal logic) is complete with respect to some class of algebras. And the proof is not difficult. In essence, one replicates the completeness proof for **K**, but works with the Lindenbaum-Tarski Algebra which satisfies the additional axiomatic constraints. As we saw earlier (recall Theorem 26) there is no general completeness result for normal modal logics with respect to frames. This is an important difference between algebraic and relational semantics.

Nonetheless, it is likely that some readers will feel a little cheated. Isn't the whole approach really just syntax in disguise? After all, algebraic semantics matches the modal language with algebraic operations that transparently mirror fundamental validities of the original logic. This does not seem like genuine semantic analysis: it has more the flavour of linking two distinct, but closely related, syntactic realms. Moreover, the algebraic satisfaction definition has a global rather than a local flavour.

This is true, but somewhat besides the point, for in spite of the general completeness result just noted, we have not yet entered the heartland of algebraic semantics. For what algebraic semantics really provides is a doorway to a larger mathematical universe. The power of algebraic semantics

comes from the wealth of ideas and techniques it enables us to bring to bear on problems in modal logic. Some of these techniques take us back, via a novel path, to the heart of relational semantics, but others take us to new territory. Let's look a little deeper.

An important theme in algebra is the *representation* of abstract mathematical structures by concrete set-theoretic structures. The point of a representation theorem is to show that some abstractly specified class of algebras picks out an intended class of concrete structures. So representation theorems are rather like completeness theorems: they show that the abstract (often equational) specification is strong enough to ensure that every abstract algebra is isomorphic to a concrete algebra. Two classic examples are Cayley's Theorem, which shows that every finite group is isomorphic to a collection of permutations, and the Stone Representation Theorem, which shows that every abstract boolean algebra is isomorphic to a field of sets (that is, a boolean closed collection of subsets of some W that contains W) with $\times$ viewed as intersection, $+$ viewed as union, and $-$ viewed as set-theoretic complement. Now, in 1952, several years before relational semantics was officially invented, Jónsson and Tarski [74, 75] proved a remarkable representation theorem for BAOs: they showed that every abstract BAO could be represented as a relational structure. Inexplicably, their paper made no mention of modal logic. This was unfortunate as their paper contained all the technical machinery needed to define *relational* semantics and prove *relational* completeness results for most commonly occurring modal logics. In essence, their result allows relational completeness proofs to be factored into an algebraic completeness step (which makes use of the Lindenbaum-Tarski Algebra) followed by a representation step (which turns this algebra into a relational structure). Nowadays, the Jónsson-Tarski Theorem is rightly considered a cornerstone of modal logic; for a detailed proof of the theorem, and examples of how to put it to work, see Chapter 6 of this handbook.

Another important theme goes under the name of *duality theory*. As we saw in Section 5, there are four key transformations on frames (disjoint unions, generated submodels, bounded morphisms, and ultrafilter extensions) and, as the Goldblatt-Thomason Theorem tells us, closure of a frame class under these model-theoretic constructions is necessary and sufficient to ensure its basic modal definability. But as we have already remarked (see Theorem 33) the original proof of the Theorem was *algebraic*. What's the algebraic connection? This: each of these four operations on frames corresponds to an operation on classes of algebras. Viewed this way, the Goldblatt-Thomason Theorem can be seen as a modal version of the Birkhoff Theorem, which identifies equationally definable classes of algebras with those classes of algebras that are closed under the formation of subalgebras, homomorphisms, and products. For a detailed discussion, we again refer the reader to Chapter 6.

But important as these two examples are, they merely hint at the wealth of techniques made available by the algebraic connection. Algebraic semantics has repeatedly proved itself a powerful analytical tool. To give another classic example, Blok [16] was able to give a detailed analysis of frame incompleteness by drawing on algebraic methods. In particular, he did so by investigating *splittings* (a concept from lattice theory) of the lattice of normal modal logics; for a discussion of Blok's work, see Chapter 7 of this handbook. Moreover, in many cases algebraic methods have been adapted to richer modal languages. A nice example is provided by the universal modality. In the algebraic setting, the universal modality allows us to define a *discriminator term*, that is, a term denoting an operator that maps 0 to 0 and all other elements to 1. Algebras with discriminator terms are particularly straightforward to work with (see Chapter 6 of this handbook) thus here algebraic semantics sheds interesting light on a relationally-natural extension of the basic modal language. But algebraic semantics also illuminates areas where relational semantics has little to say. For example, it turns out that the boolean structure of the

underlying algebras is not particularly significant. That is, it is possible to analyse modalities algebraically even if we *don't* have full classical propositional logic at our disposal. Such logics can be important in various settings, and relational semantics at present offers little in the way of insight. For further remarks and references on this application of algebraic semantics, see Chapter 6 of this handbook.

7.2 Neighbourhood semantics

For some applications, relational semantics is too strong. For example, $\Diamond(\varphi \vee \psi) \to (\Diamond\varphi \vee \Diamond\psi)$ is valid under relational semantics. But if we read $\Diamond\varphi$ as making the game-theoretic assertion that the player has a strategy forcing the outcome to satisfy φ, we might be inclined to reject it: why should possession of a strategy for a disjunction imply possession of a strategy for one of the disjuncts? For example, suppose we play a game with the following moves: you have the right to decide whether we go to a movie or a concert, and I can decide which particular movie or concert we go to. Suppose the movie I want to see is *Crash*, and that my favourite music is *Mozart*. It follows that I can force *Crash* $\vee$ *Mozart*, but (because it's you who determines the movie/concert option) I can't determine which of these two options will actually take place. Similarly, if we interpret $\Box\varphi$ epistemically we have further grounds for objection. For a start, relational semantics validates the following principle:

$$\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi).$$

Moreover, it validates the following pattern of inference: if $\models \varphi$ then $\models \Box\varphi$. These work together to enforce a strong form of logical omniscience: if an agent knows φ, then she knows all its logical consequences.

Such considerations have led to a search for weaker semantics. Perhaps the best known of these is neighbourhood semantics (introduced in Montague [96, 97] and Scott [112] and explored in Segerberg [113]). The key idea of neighbourhood semantics has a topological flavour: each point w in a model is associated with a collection of subsets of the domain (the neighbourhoods of w) and a formula of the form $\Box\varphi$ is true at w iff the set of points in a model satisfying φ is a neighbourhood of w. Let's make this precise. A neighbourhood model is a triple (W, R, V) where W is a non-empty set of states, V is a valuation, and R relates points $w \in W$ to subsets of W (that is, $R \subseteq W \times 2^W$). For any $w \in W$, let N_w be $\{U \subseteq W \mid wRU\}$; we call N_w the set of neighbourhoods of w. We interpret boxed formulas as follows:

$$\mathfrak{M}, w \models \Box\varphi \text{ iff } \{u \in W \mid \mathfrak{M}, u \models \varphi\} \in N_w,$$

and use the dual definition for diamonds:

$$\mathfrak{M}, w \models \Diamond\varphi \text{ iff } \{u \in W \mid \mathfrak{M}, u \not\models \varphi\} \notin N_w.$$

Neighbourhood semantics is a generalisation of relational semantics. To see this, note that given any relational model $\mathfrak{M} = (W, R, V)$ we can form a neighbourhood model $\mathfrak{M}^n = (W, R^n, V)$ by stipulating, for each $w \in W$ and $U \subseteq W$, that $R^n wU$ iff $U = \{u \in W \mid Rwu\}$. That is, for each $w \in W$, N_w is the singleton set containing the set of points that are R-accessible from w. Hence, for all $w \in W$ and all basic modal formulas φ, we have that $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M}^n, w \models \varphi$. In short, we can turn any relational model into an equivalent neighbourhood model.

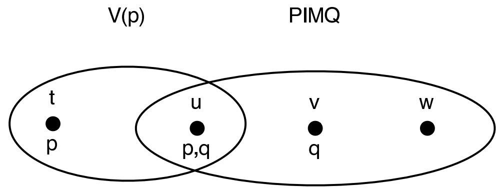


Figure 22. Neighbourhood model that falsifies $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$ at u.

But we cannot do the reverse. Consider a model $\mathfrak{M} = (W, R, V)$ such that $W = \{t, u, v, w\}$, $V(p) = \{t, u\}$ and $V(q) = \{u, v\}$, and $N_u = \{V(p), \mathit{PIMQ}\}$, where $\mathit{PIMQ} = \{u, v, w\}$. Such a model is shown in Figure 22; note that *PIMQ* is the set of points where $p \to q$ is true. Hence $\mathfrak{M}, u \models \Box(p \to q)$, as $\mathit{PIMQ} \in N_u$. Furthermore, $\mathfrak{M}, u \models \Box p$, as $V(p) \in N_u$. However $\mathfrak{M}, u \not\models \Box q$, for $V(q) \notin N_u$. So $\mathfrak{M}, u \not\models \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$. As this formula is valid under relational semantics, no relational model equivalent to $\mathfrak{M}$ exists.

Moreover, the inferential principle characteristic of relational semantics (if $\models \varphi$ then $\models \Box\varphi$) no longer holds. To see this, it suffices to consider a model $\mathfrak{M}$ consisting of a single point w such that $N_w = \emptyset$. Then $\mathfrak{M}, w \models \top$, but $\mathfrak{M}, w \not\models \Box\top$. In fact, all that remains in neighbourhood semantics is the weaker principle: if $\models \varphi \leftrightarrow \psi$ then $\models \Box\varphi \leftrightarrow \Box\psi$. Thus neighbourhood semantics does not enforce logical omniscience.

Neighbourhood semantics has been criticised as under-motivated. It may banish the spectre of logical omniscience, but does it do so in a principled way? After all, isn't there something stipulative, indeed ad-hoc, about simply asserting that certain subsets and not others are in the neighbourhood of a given point? There is a grain of truth in such criticisms, nonetheless we should not be too quick to dismiss the approach. For some applications, asserting that certain neighbouring regions are important is probably the best we can do in the way of semantic analysis. Furthermore, like relational semantics, neighbourhood semantics offers an entire *framework* for semantics; imposing further restrictions on neighbourhoods (for example, demanding that neighbourhoods be superset closed) is a mechanism which permits finer-grained semantic analyses to be attempted. See Chellas [24] for an introduction to some of the options here.

Neighbourhood semantics has some pleasant properties. For a start (if NP $\neq PSPACE$, the standard assumption) it is better behaved computationally than relational semantics:

THEOREM 55. *The satisfiability problem for neighbourhood semantics is NP-complete.*

Proof. See Vardi [140]. The key observation is that if a formula φ is satisfiable in a neighbourhood model, then it is satisfied in a model with at most $|\varphi|^2$ states, where $|\varphi|$ is the number of symbols in φ. ❑

Moreover, neighbourhood semantics meshes well with the algebraic and co-algebraic approaches discussed in Chapter 6 of this handbook.

7.3 Topological semantics

Topological semantics is one of the oldest modal semantics, and the first in which deep technical results were proved. In 1938, Tarski [122] showed that **S4** (the logic which in relational semantics is complete with respect to transitive and reflexive frames) is complete with respect to

topological spaces. Then, in 1944, McKinsey and Tarski [95] showed that **S4** is the modal logic of the real numbers, and indeed of any metric separable space without isolated points. Since this pioneering work, topological semantics has been deeply (if somewhat sporadically) studied, and many interesting results have been proved (see for example Esakia [38] and Shehtman [115]) but for many years it was rather isolated from the modal mainstream. More recently, however, partly because of the growing interest in logics of space, there has been a revival of interest. For an overview of developments in topological semantics since the time of Tarski, see Chapter 16 of this handbook; here we will introduce its basic ideas in a way that emphasises connections with our account of relational semantics. Our discussion is based on Aiello, van Benthem, and Bezhanishvili [2].

A *topological space* is a pair (W, τ), where W (the *domain*) is a non-empty set and τ (the *topology*) is a collection of subsets of W that contains both $\emptyset$ and W, is closed under finite intersections (that is, if $O, O' \in \tau$ then $O \cap O' \in \tau$) and closed under arbitrary unions (if $\{O_i\}_{i \in I} \in \tau$ then $\bigcup_{i \in I} O_i \in \tau$). A topology τ such that $\tau = 2^W$ is called *discrete*, and a topology such that $\tau = \{\emptyset, W\}$ is called *trivial*. If (W, τ) is a topological space and $O \in \tau$ then O is called an *open set*. If w is a point in an open set O, then O is called an *open neighbourhood* of w. A *closed set* is the complement of an open set.

A *topological model* is a triple $\mathfrak{M} = (W, \tau, V)$ where (W, τ) is a topological space and V is a valuation (in the sense familiar from relational semantics). We interpret proposition symbols and booleans in the usual way, but what about the modalities? Boxed formulas are handled as follows:

$$\mathfrak{M}, w \models \Box\varphi \text{ iff } (\exists O \in \tau)(w \in O \text{ and } (\forall u \in O)(\mathfrak{M}, u \models \varphi)).$$

That is, $\Box\varphi$ is true at w iff it is true at all the points of some open neighbourhood of w. Diamonds are handled dually:

$$\mathfrak{M}, w \models \Diamond\varphi \text{ iff } (\forall O \in \tau)(w \in O \text{ implies } (\exists u \in O)(\mathfrak{M}, u \models \varphi)).$$

That is, $\Diamond\varphi$ is true at w iff it is true at some point in each open neighbourhood of w.

At first blush, this looks very different from relational semantics. And there *are* some obvious semantic differences. For example, the characteristic axioms of **S4**, namely $\Box p \to p$ and $\Box p \to \Box\Box p$, are valid on all topological models, so the minimal logic is stronger than in relational semantics. But a closer look reveals the similarities. For a start, like relational semantics, topological semantics is local: the truth value of a formula at a point only depends on what happens inside the open neighbourhoods of that point. More precisely, suppose that w is a point in a topological model $\mathfrak{M}$, and that O is an open neighbourhood of w. Let $\mathfrak{M}|O$ be the model with domain O whose open sets are all the open subsets of O in $\mathfrak{M}$, and whose valuation is the restriction of the valuation V of $\mathfrak{M}$ to O (that is $V|O(p) = V(p) \cap O$). Then a simple induction shows that for all basic modal formula φ, and all points $w \in O$, $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M}|O, w \models \varphi$. Nor is it hard to find other similarities. For example, the fact that **S4** has the finite model property with respect to relational semantics is neatly matched by the fact that the basic modal language has the finite model property with respect to topological semantics.

But the similarities run deeper than these examples might suggest. In particular, topological semantics gives rise to a natural notion of bisimulation:

DEFINITION 56 (Topo-bisimulation). A topo-bisimulation between two topological models $\mathfrak{M} = (W, \tau, V)$ and $\mathfrak{M}' = (W', \tau', V')$ is a non-empty binary relation E between their domains (that is, $E \subseteq W \times W'$) such that whenever wEw' we have that:

Atomic harmony: w and w' satisfy the same proposition symbols,

Zig: if $w \in O \in \tau$, then there is an open set $O' \in \tau'$ such that $w' \in O'$ and $(\forall u' \in O')(\exists u \in O)(uEu')$, and

Zag: if $w' \in O' \in \tau'$, then there is an open set $O \in \tau$ such that $w \in O$ and $(\forall u \in O)(\exists u' \in O')(uEu')$.

If there is a topo-bisimulation between two topological models $\mathfrak{M}$ and $\mathfrak{N}$, then we say that $\mathfrak{M}$ and $\mathfrak{N}$ are topo-bisimilar. Moreover, we say that two states are topo-bisimilar if they are related by some topo-bisimulation.

Let's restate the zig clause informally: it says that for two points w and w' to be topo-bisimilar, then for any open neighbourhood O of w it must be possible to find an open neighbourhood O' of w' such that every point u' in O' is topo-bisimilar to some u in O. Figure 23 illustrates this idea (the dotted line connecting u and u' needs to be interpreted universally: *every* u' is linked to some u).

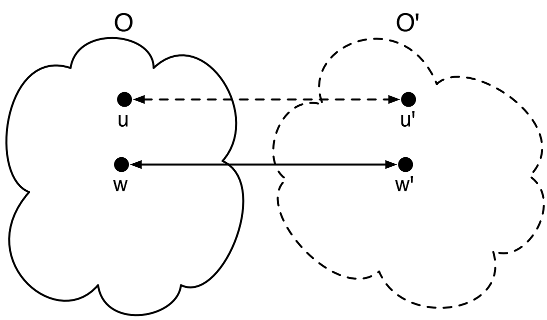


Figure 23. Zig (and zag) for topo-bisimulations

Such bisimulations are topologically natural. Two basic concepts of topology are *open maps* and *continuous maps*. For any topological spaces (W, τ) and (W', τ'), a function f from W to W' is called open if for all $O \in \tau$ we have that $f(O) \in \tau'$, and it is called continuous if for all $O' \in \tau'$ we have that $f^{-1}(O') \in \tau$. It is easy to see that every open and continuous map induces topo-bisimulations: given a valuation on one space, take its image in the other, and the resulting models are topo-bisimilar. But topo-bisimulations are also modally natural. For a start, we have the following analog of Lemma 9:

LEMMA 57 (Topo-bisimulation Invariance Lemma). *If E is a topo-bisimulation between $\mathfrak{M} = (W, \tau, V)$ and $\mathfrak{M}' = (W', \tau', V')$, and wEw', then w and w' satisfy the same basic modal formulas.*

Proof. A routine induction. ❑

As a simple illustration, we noted above that $\mathfrak{M}$ and $\mathfrak{M}|O$ (the localisation of $\mathfrak{M}$ to some open set O) were equivalent. But this is unsurprising. The identity relation between the domains of the two models is a topo-bisimulation, hence the result is a special case of this lemma.

What about the converse? Characterisation results for the general case are tricky to state (we would need to discuss what a suitable correspondence language for topological semantics is, and this would take us too far afield). But we *do* have an analog of Proposition 11:

PROPOSITION 58. *If points w and w' from two finite topological models $\mathfrak{M}$ and $\mathfrak{N}$ satisfy the same modal formulas, then there is a topo-bisimulation E between $\mathfrak{M}$ and $\mathfrak{N}$ such that wEw'.*

So far so good. But just how expressive is the basic modal language in the new setting? To pose the question a little more forcefully: what (interesting) topological conditions can the basic modal language enforce via the concept of validity? Here's one example. The formula

$$p \leftrightarrow \Box p$$

is valid on a topological model iff that model bears the discrete topology (that is, iff every subset of the domain is open). This is pleasant, but many fundamental properties lie beyond the reach of the basic language. For example, a topological space (W, τ) is *connected* iff the only elements of τ that are both open and closed are W and $\emptyset$. But this condition is not basic modal definable. For suppose for the sake of a contradiction that some formula φ does define connectedness. Consider the topological space with domain $\{1, 2\}$ under the discrete topology; this space is *not* connected as $\{1\}$ and $\{2\}$ are both open and closed. Hence we can define a model $\mathfrak{M}$ on this space that will falsify φ at some point, say 1. But then $\mathfrak{M}|\{1\}$ will falsify φ at 1 too, as $\mathfrak{M}$ and $\mathfrak{M}|\{1\}$ are topo-bisimilar. But $\mathfrak{M}|\{1\}$ bears the trivial topology, hence it is a connected space, so it should validate φ. We conclude that connectedness is undefinable.

All in all, the basic modal language turns out to be disappointingly weak when it comes to standard topological conditions. But then why stick with the basic modal language? As readers of this chapter are well aware, there are interesting ways of augmenting modal expressivity, and recently these have begun to be explored in the topological setting. For example, Shehtman [116] and Aiello and van Benthem [1] observe that connectivity becomes definable when the universal modality is added to the language:

$$A(\Diamond p \to \Box p) \to (Ap \vee A\neg p).$$

And Gabelaia notes that the T_0 condition (for any two points x and y there exists either an open neighbourhood O_x of x such that $y \notin O_x$ or an open neighbourhood O_y of y such that $x \notin O_y$) is definable in the basic hybrid language by

$$@_i\neg j \to (@_j\Box\neg i \vee @_i\Box\neg j),$$

and that the T_1 condition (every singleton set is closed) is definable by

$$i \leftrightarrow \Diamond i.$$

Gabelaia [54] proves an analog of the Goldblatt-Thomason Theorem for the basic modal language with respect to topological semantics, and Sustretov [121] has extended the result to the basic hybrid language enriched with the universal modality. However Sustretov also shows that the T_2 condition (every distinct pair of points is contained in disjoint open neighbourhoods) is not definable in this richer language.

8 MODAL LOGIC AND ITS CHANGING ENVIRONMENT

Traditional motivations for and applications of modal logic came from philosophy, and dealt with such topics as modality, knowledge, conditionals, and obligations. Other strands dealt with more mathematical topics, leading to modal logics of time, space, or provability. As time went by,

additional influences made modal logic even more diverse. Sources included computer science (for modal logics of computation and general processes), Artificial Intelligence (for modal logics for knowledge representation, non-monotonic reasoning, and belief revision), linguistics (for modal logics of grammatical structure), and the internet (for modal logics of trees). This web of new interfaces is still growing. Modern computer science, with its emphasis on new information carriers and networks of intelligent computing agents, also brings in modal logics of image processing, agency and security. And the empirical social sciences are joining in too, witness current applications of modal logic in economic game theory, or for modeling the powers of agents in social choice theory.

In the face of this diversity, the resilience of relational semantics is quite remarkable. Although nearly half a century old, its central ideas remain applicable, and applicable even when we enrich our conception of what a modal logic actually is. But what *are* the central ideas of relational semantics? In essence, this chapter has tried to make the following point clear: during the 50 or so years that relational semantics has existed, our understanding of it has become both broader and deeper. Originally conceived as a way of distinguishing and characterising logics (via soundness and completeness theorems) modal logicians have gradually unearthed the deeper mathematical themes that lie behind the seemingly modest facade of relational semantics; themes such as expressivity at the level of models versus the level of frames, the importance of bisimulation and other game-like constructions, the systematic links between the modal universe and many varieties of classical logic, ranging from first-order logic, through second-order logic, to the farther reaches of infinitary logic. Turning this perceived semantic unity into theorems is not always easy; work on combined modal logic still tends to be heavy on negative results, and first-order modal logic remains difficult territory. But unifying themes, such as guarding, and the possibility of applying ideas from abstract model theory, have emerged.

Indeed, we are tempted to conclude by playing devil's advocate: even the alternative semantics we have encountered indicate that something semantically central lies at the heart of relational semantics. For example, the Jónsson-Tarski Theorem reveals that relational semantics has an important algebraic core, and our excursion to the land of topological semantics revealed the centrality of the concept of bisimulation. Prediction is always a dangerous game (especially when it is about the future) but we believe that the interplay between theory and practice that has characterised research on modal logic throughout its history will continue to deepen our understanding of its semantic core. And, forced to place our bets, we would probably say: modal logics of games (see Chapter 20 of this handbook) will be a deep source of further insight, as will the co-algebraic semantics (discussed in Chapter 6).

ACKNOWLEDGEMENTS

We are extremely grateful to Carlos Areces for his painstaking editorial advice, to Aleksander Chagrov for his detailed list of corrections and recommendations, to Eric Kow for his patient work on the diagrams and formatting, to Sara Hobin for her meticulous proofreading and assistance with the cross-referencing, and to Dmitry Sustretov for checking Section 7. Special thanks to Daniel Gorín, Sergio Mera, Diego Figueira, and Santiago Figueira (the Buenos Aires crew) for giving the chapter its final trial.

BIBLIOGRAPHY

[1] M. Aiello and J. van Benthem. Logical patterns in space. In *Words, Proofs and Diagrams*, pages 5–25. CSLI Publications, 2002.

[2] M. Aiello, J. van Benthem, and G. Bezhanishvili. Reasoning about space: The modal way. *Journal of Logic and Computation*, 13(6):889–929, 2003.

[3] S. Akers. Binary decision diagrams. *IEEE Transactions on Computers*, C-27(6):509–516, 1978.

[4] H. Andréka, I. Hodkinson, and I. Németi. Finite algebras of relations are representable on finite sets. *Journal of Symbolic Logic*, 64:243–267, 1999.

[5] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217–274, 1998.

[6] C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation and complexity. *Journal of Symbolic Logic*, 66(3):977–1010, 2001.

[7] C. Areces, P. Blackburn, and M. Marx. Repairing the interpolation theorem in quantified modal logic. *Annals of Pure and Applied Logics*, 123(1–3):287–299, 2003.

[8] A. Baltag. STS: a structural theory of sets. In M. Zakharyaschev, K. Segerberg, M. de Rijke, and H. Wansing, editors, *Advances in Modal Logic, Volume 2*. CSLI Publications, 2000.

[9] J. Barwise and S. Feferman. *Model-Theoretic Logics*. Springer, 1985.

[10] J. Barwise and L. Moss. *Vicious Circles*. CSLI Publications, 1996.

[11] J. Barwise and J. van Benthem. Interpolation, preservation, and pebble games. *Journal of Symbolic Logic*, 29:881–903, 1999.

[12] R. Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, 66(72), 1966.

[13] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[14] P. Blackburn and M. Marx. Tableaux for quantified hybrid logic. In U. Egly and C. Fernmüller, editors, *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX 2002, Copenhagen, Denmark, July/August 2002, Proceedings*, volume 2381 of *LNAI*, pages 38–52. Springer, 2002.

[15] P. Blackburn and M. Marx. Constructive interpolation in hybrid logic. *Journal of Symbolic Logic*, 68(2):463–480, 2003.

[16] W. Blok. The lattice of modal algebras: An algebraic investigation. *Journal of Symbolic Logic*, 45:221–236, 1980.

[17] R. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Compilers*, C-35(8), 1986.

[18] J. Burch, J. Clarke, K. McMillan, D. Dill, and J. Hwang. Symbolic model checking: 10^{20} states and beyond. In *IEEE Symposium on Logic in Computer Science*, 1990.

[19] J. Burgess. Quick completeness proofs for some logics of conditionals. *Notre Dame Journal of Formal Logic*, 22:76–84, 1979.

[20] S. Buss. Bounded arithmetic and propositional proof complexity. In H. Schwichtenberg, editor, *Logic and Computation*, pages 67–122. Springer, 1997.

[21] L. Chagrova. An undecidable problem in correspondence theory. *Journal of Symbolic Logic*, 56:1261–1272, 1991.

[22] A. Chandra and P. Merlin. Optimal implementation of conjunctive queries in relational databases. In *Proceedings 9th ACM STOC*, pages 77–90, 1977.

[23] C. Chang and H. Keisler. *Model Theory*. North-Holland Publishing Company, Amsterdam, 1973.

[24] B. Chellas. *Modal Logic, an Introduction*. Cambridge University Press, 1980.

[25] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

[26] G. Corsi and S. Ghilardi. Semantical aspects of quantified modal logic. In *Knowledge, Belief, and Strategic Action*, pages 167–195. Cambridge University Press, 1992.

[27] M. Cresswell. Incompleteness and the Barcan formula. *Journal of Philosophical Logic*, 24:379–403, 1995.

[28] G. De Giacomo and M. Lenzerini. TBox and ABox reasoning in expressive description logics. In L. Aiello, J. Doyle, and S. Shapiro, editors, *KR'96: Principles of Knowledge Representation and Reasoning*, pages 316–327. Morgan Kaufmann, San Francisco, California, 1996.

[29] M. de Rijke. The modal logic of inequality. *Journal of Symbolic Logic*, 57(2):566–584, 1992.

[30] M. de Rijke. *Extending Modal Logic*. PhD thesis, ILLC, University of Amsterdam, 1993.

[31] M. de Rijke. A Lindström theorem for modal logic. In A. Ponse, M. de Rijke, and Y. Venema, editors, *Modal Logic and Process Algebra: A Bisimulation Perspective*, volume 53 of *Lecture Notes*, pages 217–230. CSLI Publications, 1995.

[32] E. Dijkstra. *A Discipline of Programming*. Prentice Hall, 1976.

[33] A. Dovier, C. Piazza, and A. Policriti. An efficient algorithm for computing bisimulation equivalence. *Theoretical Computer Science*, 311(1-3):221–256, 2004.

[34] M. Dummett and E. Lemmon. Modal logics between S4 and S5. *Zeitschrift für mathemathische Logik und Grundlagen der Mathematik*, 5:250–264, 1959.

[35] H. Ebbinghaus and J. Flum. *Finite Model Theory*. Perspectives in Mathematical Logic. Springer, 1995.

[36] E. Emerson and R. Jutla. The complexity of tree automata and logics of programs. *SIAM Journal on Computing*, 29:132–158, 1999.

[37] H. Enderton. *A Mathematical Introduction to Logic*. Academic Press, New York and London, second edition, 2001.
[38] L. Esakia. Diagonal constructions, Löb's formula, and Cantor's scattered spaces. In *Studies in Logic and Semantics*, pages 128–143. Metsniereba, 1981. In Russian.
[39] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. The MIT Press, 1995.
[40] S. Feferman. Persistent and invariant formulas for outer extensions. *Compositio Mathematica*, 20:29–52, 1968.
[41] S. Feferman and Kreisel G. Persistent and invariant formulas relative to theories of higher order. *Bulletin of the American Mathematical Society*, 72:480–485, 1966.
[42] K. Fine. Propositional quantifiers in modal logic. *Theoria*, 36:331–346, 1970.
[43] K. Fine. An incomplete logic containing $S4$. *Theoria*, 40:23–29, 1974.
[44] K. Fine. Some connections between elementary and modal logic. In S. Kanger, editor, *Proceedings of the Third Scandinavian Logic Symposium. Uppsala 1973.* North-Holland Publishing Company, 1975.
[45] K. Fine. Failures of the interpolation lemma in quantified modal logic. *Journal of Symbolic Logic*, 42(2):201–206, 1979.
[46] K. Fine and G. Schurz. Transfer theorems for stratified modal logics. In J. Copeland, editor, *Logic and Reality, Essays in Pure and Logic. In memory of Arthur Prior*, pages 169–213. Oxford University Press, 1996.
[47] M. Fischer and R. Ladner. Propositional modal logic of programs. In *Proceedings 9th ACM Symposium on Theory of Computation*, pages 286–294, 1977.
[48] M. Fischer and R. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18:194–211, 1979.
[49] M. Fitting and R. Mendelsohn. *First-Order Modal Logic*. Kluwer Academic Publishers, 1998.
[50] N. Friedman and J. Halpern. On the complexity of conditional logic. In *Proceedings of the 4th International Conference on Principles of Knowledge Representation (KR '94)*, pages 202–213, 1994.
[51] D. Gabbay. An irreflexivity lemma with applications to axiomatizations of conditions on linear frames. In U. Mönnich, editor, *Aspects of Philosophical Logic*, pages 67–89. Reidel, 1981.
[52] D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*. Oxford University Press, 1994.
[53] D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal analysis of fairness. In *Proceedings 7th ACM Symposium on Principles of Programming Languages*, pages 163–173, 1980.
[54] D. Gabelaia. Modal definability in topology. Master's thesis, University of Amsterdam, 2001.
[55] P. Gärdenfors and H. Rott. Belief revision. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, number 4, pages 35–132. Clarendon Press, 1995.
[56] S. Ghilardi. Incompleteness results in Kripke semantics. *Journal of Symbolic Logic*, 56:517–538, 1991.
[57] R. Goldblatt. Metamathematics of modal logic I. *Reports on Mathematical Logic*, 6:41–78, 1976.
[58] R. Goldblatt. Metamathematics of modal logic II. *Reports on Mathematical Logic*, 7:21–52, 1976.
[59] R. Goldblatt. Mathematical modal logic: a view of its evolution. *Journal of Applied Logic*, 1:309–392, 2003.
[60] R. Goldblatt and S. Thomason. Axiomatic classes in propositional modal logic. In J. Crossley, editor, *Algebra and Logic*, pages 163–173. Springer, 1974.
[61] V. Goranko and S. Passy. Using the universal modality: Gains and questions. *Journal of Logic and Computation*, 2:5–30, 1992.
[62] E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64:1719–1742, 1999.
[63] E. Grädel and I. Walukiewicz. Guarded Fixed Point Logic. In *Proceedings of 14th IEEE Symposium on Logic in Computer Science LICS '99, Trento*, pages 45–54, 1999.
[64] J. Halpern and M. Vardi. The complexity of reasoning about knowledge and time, I: lower bounds. *Journal of Computer and Systems Sciences*, 38(1):195–237, 1989.
[65] E. Hemaspaandra. The price of universality. *Notre Dame Journal of Formal Logic*, 37:174–203, 1996.
[66] J. Hintikka. *Knowledge and Belief.* Cornell University Press, 1962.
[67] C. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12:567–580, 1969.
[68] I. Hodkinson. Loosely guarded fragment of first-order logic has the finite model property. *Studia Logica*, 70:205–240, 2002.
[69] I. Hodkinson and M. Otto. Finite conformal hypergraph covers and Gaifman cliques in finite structures. *Bulletin of Symbolic Logic*, 9:387–405, 2003.
[70] E. Hoogland and M. Marx. Interpolation and definability in guarded fragments. *Studia Logica*, 70:373–409, 2002.
[71] E. Hoogland, M. Marx, and M. Otto. Beth definability for the guarded fragment. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, *Logic for Programming and Automated Reasoning, 6th International Conference LPAR99, Tbilisi, Georgia*, volume 1705 of *LNAI*, pages 273–285. Springer, 1999.
[72] M. Huth and M. Ryan. *Logic in Computer Science*. Cambridge University Press, 2nd edition, 1994.
[73] D. Janin and I. Walukiewicz. On the expressive completeness of the propositional mu-calculus with respect to second order logic. In *Proceedings CONCUR '96*, volume 1119 of *Lecture Notes in Computer Science*, pages 263–277. Springer, 1996.
[74] B. Jónsson and A. Tarski. Boolean algebras with operators, Part I. *American Journal of Mathematics*, 73:891–939, 1952.
[75] B. Jónsson and A. Tarski. Boolean algebras with operators, Part II. *American Journal of Mathematics*, 74:127–162, 1952.

[76] H. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, 1968.

[77] S. Kanger. The morning star paradox. *Theoria*, pages 1–11, 1957.

[78] S. Kanger. *Provability in Logic*. Almqvist & Wiksell, 1957.

[79] B. Knaster. Un théorème sur les fonctions d'ensembles. *Annales de la Société Polonaise de Mathématiques*, 6:133–134, 1928.

[80] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[81] D. Kozen and R. Parikh. A decision procedure for the propositional mu-calculus. In *Proceedings of the 2nd Workshop on Logic of Programs*, volume 164 of *LNCS*, pages 313–325. Springer, 1983.

[82] M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal Logics. *Journal of Symbolic Logic*, 56:1469–1485, 1991.

[83] S. Kripke. A completeness theorem in modal logic. *Journal of Symbolic Logic*, 24:1–14, 1959.

[84] S. Kripke. Semantic analysis of modal logic I, normal propositional calculi. *Zeitschrift für mathemathische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.

[85] S. Kripke. Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94, 1963.

[86] S. Kripke. Naming and necessity. In *Semantics of Natural Language*. Reidel, 1972.

[87] N. Kurtonina and M. de Rijke. Bisimulations for temporal logic. *Journal of Logic, Language and Information*, 6:403–425, 1997.

[88] C. Lee. Representation of switching circuits by binary-decision programs. *Bell System Technical Journal*, 38:985–999, 1959.

[89] A. Lévy. *A Hierarchy of Formulas in Set Theory*, volume 57 of *Memoirs of the American Mathematical Society*. 1965.

[90] D. Lewis. *Counterfactuals*. Blackwell, 1973.

[91] P. Lindström. On extensions of elementary logic. *Theoria*, 35:1–11, 1969.

[92] D. Makinson. Some embedding theorems for modal logic. *Notre Dame Journal of Formal Logic*, pages 252–254, 1971.

[93] M. Marx. Tolerance logic. *Journal of Logic, Language and Information*, 6:353–373, 2001.

[94] M. Marx and Y. Venema. *Multidimensional Modal Logic*, volume 4 of *Applied Logic Series*. Kluwer Academic Publishers, 1997.

[95] J. McKinsey and A. Tarski. The algebra of topology. *Annals of Mathematics*, 45:141–191, 1944.

[96] R. Montague. Pragmatics. In R. Klibansky, editor, *Contemporary Philosophy: a Survey*, pages 102–122. Florence, La Nuova Italia Editrice, 1968.

[97] R. Montague. Universal grammar. *Theoria*, 36:373–398, 1970.

[98] R. Moore. A formal theory of knowledge and action. In J. Hobbs and R. Moore, editors, *Formal Theories of the Commonsense World*, pages 319–358. Ablex, Norwood, New Jersey, 1985.

[99] M. Otto. Elementary proof of the van Benthem-Rosen characterisation theorem. Technical Report 2342, Fachbereich Mathematik, Technische Universität Darmstadt, 2004.

[100] D. Park. Concurrency and automata on infinite sequences. In *Proceedings 5th GI Conference*, pages 167–183. Springer, 1981.

[101] S. Passy and T. Tinchev. An essay in combinatory dynamic logic. *Information and Computation*, 93(2):263–332, 1991.

[102] V. Pratt. Semantical considerations on Floyd-Hoare logic. In *Proceedings 17th IEEE Symposium on Computer Science*, pages 109–121, 1976.

[103] V. Pratt. Models of program logics. In *Proceedings 20th IEEE Symposium on Foundations of Computer Science*, pages 115–222, 1979.

[104] A. Prior. *Past, Present and Future*. Clarendon Press, Oxford, 1967.

[105] A. Prior. *Papers on Time and Tense*. Oxford University Press, New edition, 2003. Edited by Hasle, Øhrstrom, Braüner, and Copeland.

[106] W. Quine. Three grades of modal involvement. In *The Ways of Paradox and Other Essays*, pages 156–174. Random House, 1953.

[107] M. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

[108] G. Restall. *An Introduction to Substructural Logics*. Routledge, 2000.

[109] E. Rosen. Modal logic over finite structures. *Journal of Logic, Language and Information*, 6:427–439, 1997.

[110] M. Ryan and P. Schobbens. Counterfactuals and updates as inverse modalities. *Journal of Logic, Language and Information*, 6:123–146, 1997.

[111] H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logic. In S. Kanger, editor, *Proceedings of the Third Scandinavian Logic Symposium. Uppsala 1973.*, pages 110–143. North-Holland Publishing Company, 1975.

[112] D. Scott. Advice on modal logic. In K. Lambert, editor, *Philosophical Problems in Logic*, pages 143–173. Reidel, 1970.

[113] K. Segerberg. *An Essay in Classical Modal Logic*. Filosofiska Studier 13. University of Uppsala, 1971.

[114] K. Segerberg. Two-dimensional modal logics. *Journal of Philosophical Logic*, 2:77–96, 1973.

[115] V. Shehtman. Modal logics of domains of the real plane. *Studia Logica*, 42:63–80, 1983.

[116] V. Shehtman. “Everywhere” and “Here”. *Journal of Applied Non-classical Logics*, 9:369–380, 1999.
[117] V. Shehtman and D. Skvortsov. Semantics of non-classical first-order predicate logics. In *Mathematical Logic*, pages 105–116. Plenum Press, New York, 1990.
[118] E. Spaan. *Complexity of Modal Logics*. PhD thesis, ILLC, University of Amsterdam, 1993.
[119] R. Street and E. Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Information and Computation*, 81:249–2644, 1989.
[120] H. Sturm. The true bisimulations for ’since’ and ’until’. *Logic and Logical Philosophy*, 10:173–183, 2002.
[121] D. Sustretov. Hybrid definability in topological spaces. Technical report, University of Amsterdam, 2005. ILLC Technical Report PP-2005-14.
[122] A. Tarski. Der Aussagenkalkül und die Topologie. *Fund. Math.*, 31:103–134, 1938.
[123] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.
[124] B. ten Cate. *Model Theory for Extended Modal Languages*. PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam, 2004.
[125] S. Thomason. An incompleteness theorem in modal logic. *Theoria*, 40:150–158, 1974.
[126] S. Thomason. Reduction of second-order logic to modal logic. *Zeitschrift für mathemathische Logik und Grundlagen der Mathematik*, 21:107–114, 1975.
[127] A. Urquhart. Decidability and the finite model property. *Journal of Philosophical Logic*, 10:367–370, 1981.
[128] J. van Benthem. *Modal Correspondence Theory*. PhD thesis, Mathematisch Instituut & Instituut voor Grondslagenonderzoek, University of Amsterdam, 1976.
[129] J. van Benthem. Two simple incomplete modal logics. *Theoria*, 44:25–37, 1978.
[130] J. van Benthem. Canonical modal logics and ultrafilter extensions. *Journal of Symbolic Logic*, 44:1–8, 1980.
[131] J. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, 1983.
[132] J. van Benthem. Beyond accessibility: Functional models for modal logic. In *Diamonds and Defaults*, pages 1–18. Kluwer, 1993.
[133] J. van Benthem. Modal frame classes revisited. *Fundamenta Informaticae*, 18:307–317, 1993.
[134] J. van Benthem. Bisimulation: the never ending story. In J. Tromp, editor, *A dynamic and quick intellect. Liber Amicorum Paul Vitányi*, pages 23–27. CWI Amsterdam, 1996.
[135] J. van Benthem. Dynamic bits and pieces. Technical Report LP-97-01, Institute for Logic, Language and Computation, University of Amsterdam, 1997.
[136] J. van Benthem. Modal foundations for predicate logic. *Bulletin of the IGPL*, 5(2):259–286, 1997.
[137] J. van Benthem. Correspondence theory. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, 2nd Edition*, number 3, pages 325–408. Kluwer Academic Publishers, 2001.
[138] J. van Benthem. Minimal predicates, fixed points and definability. *Journal of Symbolic Logic*, 70:696–712, 2005.
[139] J. van Benthem. A new modal Lindström theorem. Technical Report PP-2006-06, Institute for Logic, Language and Computation, University of Amsterdam, 2006.
[140] M. Vardi. On the complexity of epistemic reasoning. In *Proceedings 4th IEEE Symposium on Logic in Computer Science*, pages 243–252, 1989.
[141] M. Vardi. On the complexity of bounded variable queries. In *Proceedings 14th ACM Symposium on Principles of Database Systems*, pages 266–276. 1995.
[142] M. Vardi. Why is modal logic so robustly decidable? In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science 31*, pages 149–184. AMS, 1997.
[143] F. Veltman. *Logics for Conditionals*. PhD thesis, University of Amsterdam, 1985.
[144] F. Wolter. Fusions of modal logics revisited. In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic, Volume 1*, pages 361–369. CSLI Publications, 1998.
[145] G. von Wright. *An Essay in Modal Logic*. North-Holland Publishing Company, 1951.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

2

MODAL PROOF THEORY

Melvin Fitting

1	Introduction	86
2	Modal Axiomatics	87
2.1	Normal Axiom Systems	88
2.2	Soundness and Completeness	89
2.3	Difficulties, and **GL**	91
2.4	Sahlqvist Formulas	93
3	Deduction, and the Deduction Theorem	93
4	Natural Deduction	95
4.1	Classical Natural Deduction	95
4.2	Modal Natural Deduction	96
5	Semantic Tableaus	99
5.1	A Classical Tableau System	99
5.2	Destructive Modal Tableaus	101
5.3	Soundness and Completeness	103
5.4	The Logic **GL**	106
5.5	Tableau Remarks	107
6	Prefixed Tableaus	108
6.1	A Prefixed System for **K**	109
6.2	Soundness and Completeness	110
6.3	Other Modal Logics	112
7	Gentzen Systems	113
7.1	Classical Propositional Sequents	114
7.2	Modal Propositional Sequents	116
8	Hypersequents	117
8.1	Hypersequents for **S5**	117
8.2	Examples	118
8.3	Soundness and Completeness	119
9	Logics of Knowledge	121
9.1	A Basic Logic of Knowledge	121
9.2	Common Knowledge	123
10	Converse	126
11	The Universal Modality and the Difference Modality	128
12	What Are the Limitations	130
13	Quantified Modal Logic	130
13.1	Syntax and Semantics	130
13.2	Constant Domain Tableaus	131
13.3	Soundness and Completeness	132
13.4	Variations	134
14	Conclusion	135

1 INTRODUCTION

We have an interest in those modal formulas that are valid, relative to some suitable notion of validity. But verifying directly that a formula meets a validity condition is generally non-constructive. In part to get around this non-constructivity, formal proof procedures have been created, using a rich variety of mechanisms. A formal proof is a finitary certificate of validity for a formula, and a proof procedure is a specification of the requirements for being a proof. A proof procedure is sound if only valid formulas have proofs—we probably would say an unsound proof procedure is simply not a proof procedure. A proof procedure is complete if all valid formulas have proofs. For modal logics, historically, proof procedures preceded semantics, so the description above is a little anachronistic. But this is not an historical account, and anyway relational semantics is now well-developed, so let us continue as if history never happened.

It will be helpful to settle some terminology first. We assume we have an infinite list of *propositional letters*, typically P, Q, ... Formulas are built up from these in the usual way. For the time being we take as primitive implication ($\supset$), falsehood ($\bot$), and necessity ($\Box$), with negation defined by $\neg X = (X \supset \bot)$, truth by $\top = \neg\bot$, disjunction by $(X \vee Y) = (\neg X \supset Y)$, conjunction by $(X \wedge Y) = \neg(X \supset \neg Y)$, equivalence by $(X \equiv Y) = ((X \supset Y) \wedge (Y \supset X))$, and possibility by $\Diamond X = \neg\Box\neg X$. We'll use X, Y, ... for arbitrary formulas.

A *normal modal logic* is a set of formulas $\mathbf{L}$ meeting the following conditions. First, $\mathbf{L}$ contains all tautologies and all instances of the formula $\Box(X \supset Y) \supset (\Box X \supset \Box Y)$. Second, $\mathbf{L}$ contains Y if it contains X and $X \supset Y$. Third, $\mathbf{L}$ contains $\Box X$ if it contains X. Fourth and finally, with each formula X, $\mathbf{L}$ also contains all substitution instances of X—the result of uniformly replacing propositional letters with more complex modal formulas.

A large variety of formal proof procedures have been created over the years. No proof procedure suffices for every normal modal logic. Well then, what about semantically determined ones? Given any collection of frames, it is not hard to see that the set of formulas valid in all of them is a normal modal logic. No proof procedure suffices for every normal modal logic determined by a class of frames. Certain families of frames meeting special mathematical conditions determine normal logics that have had applications, and these have been given standard names—the same names are commonly used for the frame families and for the normal modal logics they determine. These normal logics tend to have proof procedures, though not every kind of proof procedure may be applicable, even to the most used of these logics. Table 1 shows the frame conditions that are most common in the literature. When traditional names are available I have employed them, but other naming conventions are in use as well. For instance, **B** is also known as **KTB**. In this chapter I will present several kinds of proof procedures, using the logics of Table 1 as examples. I will not attempt to say, for each proof procedure, exactly what range of logics it is good for. Such things are often difficult to determine. But some proof procedures apply to a fairly broad range of normal logics, others to a narrower range. Some provide proofs that humans find intuitively appealing, others are better for machine implementation. I merely wish to display something of the variety available.

Name	Frame Condition
K	none
T	reflexive
K4	transitive
S4	reflexive, transitive
KB	symmetric
B	reflexive, symmetric
S5	reflexive, transitive, symmetric
D	serial
KD4	serial, transitive

Table 1. Some Frame Families for Normal Modal Logics

2 MODAL AXIOMATICS

Axiomatic proof procedures are perhaps the easiest to explain to people. Rules are simple to state and motivate. Candidates for proofs are easily checked for correctness. Unfortunately, axiomatic proofs are generally hard to discover. Today, when automatibility of proof procedures is an important concern, axiomatic systems receive increasingly short shrift. Nonetheless, axiomatic characterizations often make it relatively easy to compare modal logics, and knowing the axioms and rules for a logic supplies a special understanding, even if one does not spend much time constructing axiomatic proofs. And there are modal logics with axiom systems but no decent automatable proof procedures. Let us begin our discussion of proof procedures with axiom systems, then.

An axiomatic proof is a sequence of formulas, each of which is from a specified collection, called *axioms*, or follows from earlier terms of the sequence by a *rule of derivation*. An axiomatic proof proves its last line, or equivalently, proves each of its lines. A proved formula is a *theorem* of the axiomatic system. Of course there is an *effectiveness* requirement—we should be able to tell whether a formula is an axiom or not, and whether a rule of inference is applicable or not. This will be obvious for the axiom systems considered here. Axiom systems differ from each other in the choice of axioms and rules of derivation. They also differ in which propositional connectives and modal operators are taken as primitive, but this is not a deep issue. Early modal axiom systems differed considerably from modern ones in their choices, but this is not an historical account. All current axiom systems for normal modal logics follow the style introduced in [31], so this will be the approach here.

Axioms are particular formulas. It is common to specify them by giving *axiom schemes*. An axiom scheme is a pattern, and any formula matching that pattern is an axiom. When axiom schemes are used, typically a proof procedure will have a finite number of axiom schemes but an infinite number of axioms. An alternative method is to specify a finite number of axioms, and adopt substitution of formulas for propositional letters as an explicit rule of inference. This tends to be more complicated, and we will follow the axiom scheme approach.

2.1 Normal Axiom Systems

Ever since [31], modal axiom systems have been *modular*–whenever possible, systems build on other ones instead of starting over. In particular, modal axiom systems (usually) build on classical propositional logic. Since classical logic is well-understood, we can skip detailed consideration of it. Among our axioms will be the following.

Classical Logic All tautologies, or at least enough of them to ensure the derivability of all the rest.

In addition we will always assume we have the following axiom scheme.

Normality Scheme All formulas of the form $\Box(X \supset Y) \supset (\Box X \supset \Box Y)$

And as rules of inference, there is the familiar *modus ponens*, plus the essentially modal *necessitation rule*, introduced by Gödel.

Modus Ponens Conclude Y from X and $X \supset Y$

Necessitation Conclude $\Box X$ from X

The Rule of Necessitation requires some comment. It does not say X is necessary if it is true—it says X is necessary if it has a proof. The idea is, things that are provable are surely necessary—they must hold under all circumstances.

It is standard to call the minimal axiom system in the sense above K, for Kripke—in fact it axiomatizes the normal modal logic **K**, as will be shown below. All other axiom systems we consider will be obtained by adding extra axioms to K. First examples of modal axiomatic proofs are almost always the same, so let us round up the usual suspects. To begin, $\Box(X \wedge Y) \supset \Box X$ is a theorem of K. Better said, any formula of this form is a theorem, but we won't be so precise from now on. Here is a proof.

1.	$(X \wedge Y) \supset X$	tautology
2.	$\Box((X \wedge Y) \supset X)$	from 1 by Necessitation
3.	$\Box((X \wedge Y) \supset X) \supset (\Box(X \wedge Y) \supset \Box X)$	Normality Scheme
4.	$\Box(X \wedge Y) \supset \Box X$	Modus Ponens on 2, 3

Having seen this, it should be easy for you to show that the following is a *derived rule* in the axiom system—that is, any proof making use of it can be expanded to a proper proof not using it.

Regularity Conclude $\Box X \supset \Box Y$ from $X \supset Y$

With Regularity, it is trivial to show that $\Box(X \wedge Y) \supset \Box Y$ is a theorem, and consequently so is $\Box(X \wedge Y) \supset (\Box X \wedge \Box Y)$, using classical reasoning. Here is an *abbreviated* proof of the converse. We thus have $\Box(X \wedge Y) \equiv (\Box X \wedge \Box Y)$.

1.	$X \supset (Y \supset (X \wedge Y))$	tautology
2.	$\Box X \supset \Box(Y \supset (X \wedge Y))$	from 1 by Regularity
3.	$\Box(Y \supset (X \wedge Y)) \supset (\Box Y \supset \Box(X \wedge Y))$	Normality Scheme
4.	$\Box X \supset (\Box Y \supset \Box(X \wedge Y))$	from 2 and 3 by classical logic
5.	$(\Box X \wedge \Box Y) \supset \Box(X \wedge Y)$	from 4 by classical logic

Other common modal axiom systems are obtained by adding axiom schemes to K. The standard names for several of these schemes are given in Table 2. If scheme T, say, is added to K, we will call the resulting axiom system T, and similarly for the other cases.

Name	Axiom Schemes
K	no additional axioms
T	$\Box X \supset X$
$K4$	$\Box X \supset \Box\Box X$
$S4$	$T + K4$
KB	$X \supset \Box\Diamond X$
B	$T + KB$
$S5$	$T + K4 + KB$
D	$\Box X \supset \Diamond X$ or $\Diamond\top$
$KD4$	$D + K4$

Table 2. Some Normal Modal Logics

Incidentally, there is a whole class of modal logics called *regular* that are weaker than the normal modal logics. They are axiomatized by replacing the Necessitation Rule by the Regularity Rule. They too have a semantics, and the usual variety of proof procedures, but we will not be considering them further here—[19] has a treatment.

2.2 Soundness and Completeness

Suppose we have a normal modal logic that is characterized semantically, as the set of formulas valid in a certain class of frames. Let us call the modal logic **L**, the frames **L**-frames, and models based on them **L**-models. So, $X \in \mathbf{L}$ if and only if X is true at every possible world of every **L**-model. And suppose we have a candidate for an axiomatization of **L**: a collection of axiom schemes, L, which we add to the axiomatic system K. How might one show soundness and completeness for the axiomatization L, relative to the class of **L**-models?

Soundness is generally a simple matter for axiom systems. One establishes that every line of a proof in L is valid in all **L**-models, and hence all theorems are valid. To do this it is enough to show all axioms are valid, and the rules preserve validity. That the rules of L (that is, of K) preserve validity is immediate. The Rule of Necessitation corresponds directly to one of the conditions for a normal modal logic, and Modus Ponens to another. All that is left is to verify the validity of the axioms. And clearly, all tautologies, and instances of the Normality Scheme, are valid in every model—this is simple to check.

So soundness comes down to the following straightforward issue: is it the case that all instances of L axiom schemes are valid in **L**-frames? It is easily checked that instances of scheme T are valid in **T**-frames, instances of $K4$ are valid in **K4**-frames, and so on. This kind of thing gives all the 'standard' soundness results—in particular, for all the axiomatic systems of Table 2 with respect to the corresponding frame classes of Table 1.

Completeness is more work—sometimes much more—but often the method of *canonical models* works uniformly and well. Suppose, as above, that L is a set of axiom schemes. We construct the canonical model $\mathcal{M}$ for L.

Call a set S of formulas *L-inconsistent* if there is a finite subset $\{X_1, \ldots, X_n\}$ of S such that $(X_1 \wedge \ldots \wedge X_n) \supset \bot$ is a theorem of L. Call S *L-consistent* if it is not L-inconsistent, and *maximally L-consistent* if it is L-consistent and has no proper extension that is L-consistent. Lindenbaum's Lemma applies, in the usual way, to say that every L-consistent set has a maximal L-consistent extension. Since the proof of Lindenbaum's Lemma is by a construction that will come up several times, in various forms, let me remind you of how it goes.

Lindenbaum Construction Suppose S is L-consistent. Enumerate the (countably many) formulas of the language, Z_1, Z_2, …, and define the following sequence of sets.

$$S_1 = S$$
$$S_{n+1} = \begin{cases} S_n \cup \{Z_n\} & \text{if } L\text{-consistent} \\ S_n & \text{otherwise} \end{cases} \tag{1}$$

One then shows that $S_1 \subseteq S_2 \subseteq S_3 \subseteq \ldots$, each S_n is L-consistent, $\cup_n S_n$ is L-consistent, and $\cup_n S_n$ is maximally L-consistent.

In fact, this construction is more general than it would first seem. Say a collection $\mathcal{C}$ of sets is of *finite character* provided $S \in \mathcal{C}$ if and only if every finite subset of S is in $\mathcal{C}$. (It is immediate from the definition above that the collection of L-consistent sets is of finite character.) The Lindenbaum construction can easily be adapted to show: if $S \in \mathcal{C}$ and $\mathcal{C}$ is of finite character, then S can be extended to a maximal member of $\mathcal{C}$. This observation makes things a little easier for us later on.

Now, let $\mathcal{M} = \langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle$ be the model constructed as follows. $\mathcal{G}$ is the set of all maximally L-consistent sets of formulas. For $w, w' \in \mathcal{G}$, $w\mathcal{R}w'$ provided $\{X \mid \Box X \in w\} \subseteq w'$. And finally, $w \in \mathcal{V}(P)$ provided $P \in w$. This is the *canonical model* for L. The chief fact concerning it is the so-called *Truth Lemma*: for every formula X and possible world $w \in \mathcal{G}$

$$X \in w \text{ if and only if } \mathcal{M}, w \Vdash X \tag{2}$$

The Truth Lemma is proved by induction on the degree of X. The atomic case is by definition, and the propositional connective cases are straightforward. Here is a sketch of the modal case. We wish to show (2) is true for $\Box Z$ under the assumption that it holds for simpler formulas, in particular, for Z. Half is simple. Suppose $\Box Z \in w$, and let w' be an arbitrary world such that $w\mathcal{R}w'$. By definition of $\mathcal{R}$, $Z \in w'$; by the induction hypothesis, $\mathcal{M}, w' \Vdash Z$; so since w' was arbitrary, $\mathcal{M}, w \Vdash \Box Z$. The other direction requires more work.

Suppose $\Box Z \notin w$. Consider the set $S = \{X \mid \Box X \in w\} \cup \{\neg Z\}$. This is L-consistent, because if not, there would be a finite subset $\{\Box X_1, \ldots, \Box X_n\}$ of w such that

1.	$(X_1 \wedge \ldots \wedge X_n \wedge \neg Z) \supset \bot$	definition of inconsistent
2.	$(X_1 \wedge \ldots \wedge X_n) \supset Z$	by classical reasoning from 1
3.	$\Box(X_1 \wedge \ldots \wedge X_n) \supset \Box Z$	Regularity on 2
4.	$(\Box X_1 \wedge \ldots \wedge \Box X_n) \supset \Box Z$	using results shown earlier

But each $\Box X_i \in w$, and it follows from the maximal L-consistency of w that $\Box Z \in w$, which is a contradiction. Thus we know that S is L-consistent. Extend it to a maximal

L-consistent set w'. By definition, $w' \in \mathcal{G}$ and clearly $w\mathcal{R}w'$. And $\neg Z \in w'$ so $Z \notin w'$ and by the induction hypothesis, $\mathcal{M}, w' \not\Vdash Z$, hence $\mathcal{M}, w \not\Vdash \Box Z$.

With the Truth Lemma established, it follows that the canonical model is a universal counter-model for L—it provides counterexamples for all non-theorems of L. For, suppose X is not a theorem of axiom system L. Then $\{\neg X\}$ is **L**-consistent, and so can be extended to a maximal L-consistent set w. w is a world of the canonical model and, since $X \notin w$, by the Truth Lemma, $\mathcal{M}, w \not\Vdash X$.

If the canonical model for L happens to be an **L**-model, this establishes completeness since, if X is not a theorem of L, there is an **L**-model in which X fails—the canonical one. Since **K** imposes no conditions on frames, we now have the completeness of axiom system K relative to **K**. As a matter of fact, the canonical model for T is a **T**-model, for $K4$ is a **K4**-model, and so on for the various entries in Tables 1 and 2. Here is a sketch of the verification for one case—$K4$. Let $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{V} \rangle$ be the canonical model for $K4$—I'll show it is transitive. Well, suppose $w_1, w_2, w_3 \in \mathcal{G}$ and $w_1\mathcal{R}w_2$ and $w_2\mathcal{R}w_3$. And say $\Box X \in w_1$. Since $\Box X \supset \Box\Box X$ is an axiom of $K4$, and possible worlds of the canonical model are maximally consistent, hence deductively closed, it follows that $\Box\Box X \in w_1$. By definition of $\mathcal{R}$ in the canonical model, $\Box X \in w_2$, and hence $X \in w_3$. It has been shown that $\{X \mid \Box X \in w_1\} \subseteq w_3$, so $w_1\mathcal{R}w_3$, and thus we have transitivity. I'll leave it to you to check the other cases. Thus in one construction we have completeness for a large class of axiom systems, relative to a large class of frame families.

2.3 Difficulties, and **GL**

One should not go away with the impression that canonical models solve all problems. There are standard axiomatically formulated logics for which completeness results can be proved relative to a class of frames, but not by a direct canonical model technique. A simple example is the well-known provability logic GL, axiomatized by adding the GL schema $\Box(\Box X \supset X) \supset \Box X$ to $K4$ (or equivalently, to K, though this takes some work to show). See [8, 9, 56] for the full story. GL is sound and complete with respect to two different classes of frames. One class, call it $\mathbf{GL}^w$, consists of transitive, well-founded frames—well-founded frames are those in which there are no infinite sequences of worlds w_1, w_2, w_3, ..., with $w_i\mathcal{R}w_{i+1}$. (Technically, it is the relation that is converse to $\mathcal{R}$ that is well-founded, but we can ignore the point here.) The other class, call it $\mathbf{GL}^f$, consists of frames that are transitive, irreflexive, and finite. For applications to arithmetic the class $\mathbf{GL}^f$ is the more interesting, but clearly one cannot prove completeness of GL with respect to $\mathbf{GL}^f$ using canonical models for the simple reason that a canonical model is infinite, and hence not a member of the designated class of frames. In this section I'll briefly sketch how the logic can be handled axiomatically. We will see it again after tableaus have been introduced.

Every $\mathbf{GL}^f$ frame is also a $\mathbf{GL}^w$ frame, so a soundness proof with respect to $\mathbf{GL}^w$ establishes soundness with respect to both. And for this it is enough to show all instances of the schema $\Box(\Box X \supset X) \supset \Box X$ are valid in $\mathbf{GL}^w$ frames. Well, suppose we had a model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{V} \rangle$, based on a $\mathbf{GL}^w$ frame, and a possible world w_1 of it, such that $\mathcal{M}, w_1 \Vdash \Box(\Box P \supset P)$ but $\mathcal{M}, w_1 \not\Vdash \Box P$. By the latter, there must be a world w_2 with $w_1\mathcal{R}w_2$ and $\mathcal{M}, w_2 \not\Vdash P$. Of course we also have $\mathcal{M}, w_2 \Vdash \Box P \supset P$. It follows that $\mathcal{M}, w_2 \not\Vdash \Box P$. Hence there exists a world w_3 with $w_2\mathcal{R}w_3$ and $\mathcal{M}, w_3 \not\Vdash P$. Since $\mathcal{R}$ is transitive, $\mathcal{M}, w_3 \Vdash \Box P \supset P$. It follows that $\mathcal{M}, w_3 \not\Vdash \Box P$. So we can repeat the

argument, getting a world w_4 accessible from w_3, at which we have $\mathcal{M}, w_4 \not\Vdash P$ and $\mathcal{M}, w_4 \Vdash \Box P \supset P$, and so on. This contradicts well-foundedness of the frame. Hence there can be no such model $\mathcal{M}$, so $\Box(\Box P \supset P) \supset \Box P$ must be valid in all $\mathbf{GL}^w$ frames.

Once again, every $\mathbf{GL}^f$ frame is also a $\mathbf{GL}^w$ frame, so a completeness proof with respect to $\mathbf{GL}^f$ will show completeness with respect to both. As noted above, a canonical model construction cannot work. But something not radically different from it does. Let Z be a formula that is not provable—I will construct a $\mathbf{GL}^f$ model, $\mathcal{M}_Z$, that invalidates it.

Define $\mathbf{sub}(Z)$ to be the set of all subformulas of Z, and negations of subformulas of Z—a finite set. Now, let $\mathcal{G}_Z$ be the collection of all *maximally GL-consistent subsets of* $\mathbf{sub}(Z)$—again a finite set. It is easy to see that any GL-consistent subset of $\mathbf{sub}(Z)$ can be extended to a maximal such set. Next I'll define an auxiliary relation (used shortly to define the actual accessibility relation): for $w, w' \in \mathcal{G}_Z$, set $w\mathcal{R}_0 w'$ if $\{X, \Box X \mid \Box X \in w\} \subseteq w'$. Now, here is the real thing: for $w, w' \in \mathcal{G}_Z$, set $w\mathcal{R}_Z w'$ if $w\mathcal{R}_0 w'$ but not $w'\mathcal{R}_0 w$. Finally, let $w \in \mathcal{V}_Z(P)$ provided $P \in w$. We thus have our model $\mathcal{M}_Z = \langle \mathcal{G}_Z, \mathcal{R}_Z, \mathcal{V}_Z \rangle$.

It is obvious that $\mathcal{G}_Z$ is finite. It is equally obvious that $\mathcal{R}_Z$ is irreflexive. If we had that $\mathcal{R}_Z$ was transitive, we would know that $\mathcal{M}_Z$ was a $\mathbf{GL}^f$ model. In fact, this is the case, but for readability I'll give the argument in a separate paragraph.

Suppose $w_1\mathcal{R}_Z w_2$ and $w_2\mathcal{R}_Z w_3$; I'll show $w_1\mathcal{R}_Z w_3$. I'll leave the key step to you: show that $\mathcal{R}_0$ is transitive. Given this, we must have $w_1\mathcal{R}_0 w_3$, since we have $w_1\mathcal{R}_0 w_2$ and $w_2\mathcal{R}_0 w_3$, and $\mathcal{R}_0$ is transitive. If we had $w_3\mathcal{R}_0 w_1$, since we have $w_1\mathcal{R}_0 w_2$ and $\mathcal{R}_0$ is transitive, we would have $w_3\mathcal{R}_0 w_2$, and we do not. Thus we do not have $w_3\mathcal{R}_0 w_1$. It follows that $w_1\mathcal{R}_Z w_3$.

Thus $\mathcal{M}_Z$ is an $\mathbf{GL}^f$ model. To show it is a counter-model to Z we need an analog of the *Truth Lemma* stated earlier as (2). The original version must be replaced with the following

$$\text{For } X \in \mathbf{sub}(Z),\ X \in w \text{ if and only if } \mathcal{M}_Z, w \Vdash X \tag{3}$$

The proof of (3) is almost the same as that of (2), by induction on the complexity of X. I'll just give one step, but it is the most significant one. Suppose that (3) is known for the formula X, $w \in \mathcal{G}_Z$, $\Box X \in \mathbf{sub}(Z)$, and $\Box X \notin w$. I'll show $\mathcal{M}_Z, w \not\Vdash \Box X$.

We are assuming $\Box X \notin w$. Let S be the set $\{Y, \Box Y \mid \Box Y \in w\} \cup \{\Box X, \neg X\}$. This is GL-consistent, because if not, there would be a finite subset $\{\Box Y_1, \ldots, \Box Y_n\}$ of w such that

1. $(Y_1 \wedge \ldots \wedge Y_n \wedge \Box Y_1 \wedge \ldots \wedge \Box Y_n \wedge \Box X \wedge \neg X) \supset \bot$
   definition of inconsistent
2. $(Y_1 \wedge \ldots \wedge Y_n \wedge \Box Y_1 \wedge \ldots \wedge \Box Y_n) \supset (\Box X \supset X)$
   by classical reasoning from 1
3. $\Box(Y_1 \wedge \ldots \wedge Y_n \wedge \Box Y_1 \wedge \ldots \wedge \Box Y_n) \supset \Box(\Box X \supset X)$
   Regularity on 2
4. $(\Box Y_1 \wedge \ldots \wedge \Box Y_n \wedge \Box\Box Y_1 \wedge \ldots \wedge \Box\Box Y_n) \supset \Box(\Box X \supset X)$
   distributing $\Box$ over $\wedge$ (Section 2.1)
5. $(\Box Y_1 \wedge \ldots \wedge \Box Y_n) \supset \Box(\Box X \supset X)$
   using the $K4$ axiom
6. $(\Box Y_1 \wedge \ldots \wedge \Box Y_n) \supset \Box X$
   using the GL axiom

But each $\Box Y_i \in w$, and so $\Box X \in w$, a contradiction. Thus we know that S is GL-consistent. Extend it to a maximal GL-consistent subset w' of $\mathbf{sub}(Z)$. By definition, $w' \in \mathcal{G}_Z$. Also by definition, $w\mathcal{R}_0 w'$. And we do not have $w'\mathcal{R}_0 w$ since $\Box X \in w'$ but $\Box X \notin w$. Thus $w\mathcal{R}_Z w'$. And $\neg X \in w'$ so $X \notin w'$ and by the induction hypothesis, $\mathcal{M}_Z, w' \not\Vdash X$, hence $\mathcal{M}_Z, w \not\Vdash \Box X$.

With (3) established, there must be a possible world in the model $\mathcal{M}$ at which the unprovable formula Z is false (a maximal consistent extension of $\{\neg Z\}$). Thus we have completeness.

A canonical model is a universal counter-model—it invalidates all unprovable formulas. The present construction, while very similar, does not produce any such thing. Each formula, Z, is invalidated in a counter-model, $\mathcal{M}_Z$, of its own.

The construction above makes use of a variant of a technique known as *filtration*. A more standard version would begin by constructing a model in which worlds are maximal consistent sets, in the usual sense, and then identifying those worlds that agree on the subformulas of Z. This is one technique among many for constructing models when the canonical construction does not work. These constructions are often ingenious, often intricate, and beyond the scope of the present chapter.

2.4 Sahlqvist Formulas

Though canonical models do not solve all completeness problems, they do for the logics considered in Section 2.2. One naturally wonders what these logics have in common that makes them so nice. In [54] a remarkable answer to this question was given—see [6] for an insightful, elegant treatment. Sahlqvist defined syntactically a class of modal formulas having two important properties. First, there is an algorithm (Sahlqvist-van Benthem) for associating with each formula of the class a first-order condition on frames. These frames will validate the corresponding modal formulas. And second, the canonical model for a logic axiomatized by Sahlqvist formulas will satisfy the first-order conditions determined by the formulas. Thus, the canonical model technique must work for modal logics whose axioms are Sahlqvist formulas.

All formulas in Table 2 are Sahlqvist formulas, and the frame conditions they determine using the Sahlqvist-van Benthem algorithm are those in Table 1. On the other hand, the GL scheme, $\Box(\Box X \supset X) \supset \Box X$ is not Sahlqvist, the frame classes $\mathbf{GL}^f$ and $\mathbf{GL}^w$ are not first-order definable, and canonical models do not work.

A full discussion of the fundamental Sahlqvist results (and their limitations) can be found elsewhere in this book, in Chapters 1 and 7, so I will say no more about them here.

3 DEDUCTION, AND THE DEDUCTION THEOREM

In many logics (classical logic is the classical example) one introduces a notion of *derivation*, or *deduction*, or *consequence*—besides what is provable, what follows from what. Typically, Y follows from a set S of formulas, *premises*, in some axiomatic system if Y becomes provable when members of S are added to the system's axioms. Then one connects deduction and provability by showing a *deduction theorem*: Y is a consequence of the set $S \cup \{X\}$ if and only if $X \supset Y$ is a consequence of S. Taking S to be empty we have the important special case: Y has a derivation from X if and only if $X \supset Y$ follows

from $\emptyset$, that is, if and only if $X \supset Y$ is a theorem of the original axiomatic system. This is an important tool, both theoretically and practically, because it allows us to prove an implication $X \supset Y$ by carrying out a derivation, of Y from X, and such a derivation is often easier to discover since we have more material to work with, namely we have X.

Modal logic raises problems for the notion of deduction. Suppose we want to show $X \supset Y$ in some modal axiom system by deriving Y from X. So we add X to our axioms. Say, to make things both concrete and intuitive, that X is "it is raining" and Y is "it is necessarily raining." Since X has been added to the axiom list the necessitation rule applies, and from X we conclude $\Box X$, that is, Y. Then the deduction theorem would allow us to conclude that if it is raining, it is necessarily raining. This does not seem right—nothing would ever be contingent. On the other hand, if we are working in the modal logic **K**, and we want to see what happens if we strengthen it to **T** by adding all instances of the scheme $\Box X \supset X$, we certainly want the necessitation rule to apply to these instances. Things are not simple.

In the examples above, instances of the axiom scheme $\Box X \supset X$ are clearly intended to be understood as logical truths, and we would expect the necessitation rule to apply to them. But "it is raining" is a contingent truth, and necessitation should not apply. In modal logics, a proper notion of deduction must allow two kinds of premises, *global*, to which the necessitation rule applies, and *local*, to which it does not. The following definition is from [21].

DEFINITION 1. Let L be a set of axiom schemes for a normal modal logic—extending K. Let S and U be sets of formulas (not schemes) and X be a single formula (also not a scheme). The formula X *is deducible from* the set S of *global* premises and the set U of *local* premises in L if there is a sequence of formulas ending with X, consisting of a global part, coming first, and a local part, coming last. In the global part each formula is an instance of a member of L, a member of S, or follows from earlier formulas by modus ponens or necessitation. In the local part each formula is an instance of a member of L, a member of U, or follows from earlier formulas by modus ponens (necessitation is not allowed in this part). If X is so deducible, this is symbolized by $S \vdash_L U \to X$.

The working content of the definition above is simple: in an axiomatic derivation of X in L one can proceed as one does in a proof, using members of S and U as additional axioms, except that once we start using members of U, the necessitation rule can no longer be applied.

Since we have two kinds of premises, we have two versions of the deduction theorem. Here they are.

THEOREM 2 (Deduction). *Let S and U be sets of formulas, X and Y be single formulas, and L be a set of axiom schemes extending K.*

1. $S \vdash_L U \cup \{X\} \to Y$ *if and only if* $S \vdash_L U \to (X \supset Y)$.
2. $S \cup \{X\} \vdash_L U \to Y$ *if and only if* $S \vdash_L U \cup \{X, \Box X, \Box^2 X, \Box^3 X, \dots\} \to Y$.

The proof of the theorem above is a variation on that of the classical deduction theorem. I'll omit it here. The significant thing is that there are two versions. In reading the literature in modal logic it is important to notice, when an author talks of deduction, whether the premises are local or global. Both versions appear, often without the local/global qualification, and this can lead to some confusion.

There is a semantic counterpart of the local/global distinction, which accounts for the terminology.

DEFINITION 3. Let **L** be a family of frames, thus characterizing a normal modal logic. Let S and U be sets of formulas (not schemes) and X be a single formula (also not a scheme). The formula X is a *semantic consequence* of the set S of *global* premises and the set U of *local* premises in **L** provided, for every **L**-model $\mathcal{M}$ in which all members of S are valid (true at every possible world), for each possible world w of $\mathcal{M}$ at which all members of U are true, X is true. This is symbolized by $S \models_{\mathbf{L}} U \to X$.

In Section 2.2 soundness and completeness issues were discussed. There is a more general notion, taking deduction into account.

DEFINITION 4. Let L be a set of axiom schemes, extending K, and let **L** be a class of frames. L is *strongly complete* (and sound) with respect to **L** provided, for all sets S and U of formulas, and for all formulas X:

$$S \vdash_L U \to X \Longleftrightarrow S \models_{\mathbf{L}} U \to X.$$

Completeness arguments using canonical models tend to actually establish strong completeness. This is the case for all the axiom systems of Table 2 relative to the corresponding classes of frames in Table 1. It is not always so straightforward, however. In Section 2.3 I mentioned axiomatically formulated GL, and a corresponding class of frames **GL**. This is an instance where completeness, but not strong completeness is the case. Strong completeness cannot be taken for granted.

4 NATURAL DEDUCTION

Someone once said that in mathematics every important theorem eventually becomes a definition. Well, the Deduction Theorem is sufficiently important that proof systems have been created with it as part of the basic machinery, rather than being a derived rule. Such systems are called *natural deduction* systems, and were originally introduced by Gentzen, [29], and Jaskowski, [40]. Prawitz wrote a classic study of these systems, [52]. Modal versions have been introduced, with those of Fitch being the best-known [16, 55]. Here I'll briefly sketch a classical and several modal natural deduction systems.

4.1 Classical Natural Deduction

Recall that our basic classical connectives are $\supset$ and $\bot$, with other connectives taken as defined. In particular, $\neg X$ is $X \supset \bot$. I'll only give rules for these, though rules for other connectives can be introduced. In a natural deduction proof, *assumptions* are made, then eventually these assumptions are *discharged* using a principle like that embodied in the Deduction Theorem. Parts of proofs involving assumption, reasoning, and assumption discharge are called *subordinate proofs*, and are characteristic of natural deduction systems. Notation differs for indicating a subordinate proof. I'll enclose it in a box. The rules are given in Table 3.

There is some variety to what is called a natural deduction system in the literature. Sometimes proofs have a tree structure, as in [52]. Here natural deduction proofs are Fitch-style, [17], in which a proof is a sequence of formulas, as in axiom systems. In

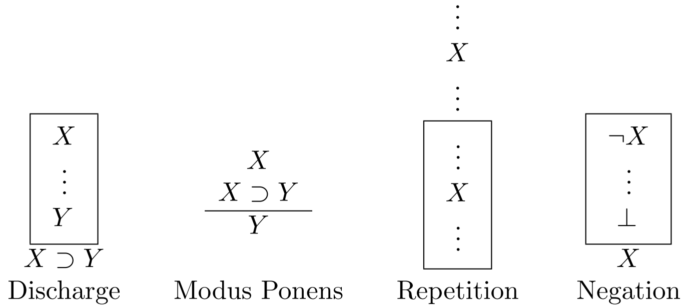


Table 3. Classical Natural Deduction Rules

addition, subordinate proofs can be started at any point—parts of a proof might be boxed, and boxes can be nested, but they cannot overlap. *The first formula in a box is understood to be an assumption—a premise.* Then the Discharge rule in Table 3 incorporates the principle of the Deduction Theorem: having assumed X as a premise, and having deduced Y, the premise X can be *discharged* and $X \supset Y$ concluded. Premise discharge is symbolized by closing off a box, thus ending a subordinate proof. Before explaining the other rules, one more notion is needed. I'll say that in a proof, two formulas, or a formula and a box, are at the *same level* if they are inside the same (nested) boxes. Modus Ponens, in Table 3, can only be applied if X and $X \supset Y$ are at the same level (though the order of the two formulas does not matter). Likewise in the Repetition rule, the upper occurrence of X must be at the same level as the box into which X is shown being repeated. A formula is a *theorem* of this natural deduction system if it is the last line of a proof and does not occur inside a box.

Figure 1 contains an example of a simple classical proof, in this system, of $(\neg Q \supset P) \supset ((\neg Q \supset \neg P) \supset Q)$. In this, 1, 2, and 4 are premises; 3 is from 1 by Repetition; 5 is from 2 by Repetition; 6 is from 4 and 5 by Modus Ponens; $6'$ is just 6 unabbreviated; 7 is from 3 by Repetition; 8 is from 4 and 7 by Modus Ponens; 9 is from 6' and 8 by Modus Ponens; 10 is by Negation; 11 and 12 are by Discharge.

If we drop the Negation rule, exactly intuitionistic implication is captured. If Negation is replaced by a rule allowing us to conclude X from $\bot$, this gives us intuitionistic negation. Other intuitionistic connectives can be captured as well, but this is too far afield for present purposes. Also, various derived rules will probably have occurred to you—for instance, Repetition can involve deeply nested boxes, instead of just going 'one box in.' Conditions were stated as they were to allow the easy addition of modal rules. I'll leave simplifications to you.

4.2 Modal Natural Deduction

Recall that whether or not the necessitation rule applied to premises in an axiomatic deduction led us to distinguish between two kinds of premises, local and global. A similar point comes up with natural deduction proofs, and we are led to create two kinds of subordinate proofs. One kind is as before, and follows the standard rules. The other kind is called a *strict* subordinate proof. I will symbolize it by enclosing it in a double-walled box. Think of a strict subordinate proof as an argument taking place in

1. $\neg Q \supset P$
2. $\neg Q \supset \neg P$
3. $\neg Q \supset P$
4. $\neg Q$
5. $\neg Q \supset \neg P$
6. $\neg P$

6′. $P \supset \bot$

7. $\neg Q \supset P$
8. P
9. $\bot$
10. Q
11. $(\neg Q \supset \neg P) \supset Q$
12. $(\neg Q \supset P) \supset ((\neg Q \supset \neg P) \supset Q)$

Figure 1. Classical Natural Deduction Proof

an arbitrary alternative world. Equivalently, think of it as an argument taking place within the scope of a necessitation operator. *A strict subordinate proof does not have an initial premise*, as ordinary subordinate proofs do, and one can be started at any point. Whatever is shown in a strict subordinate proof has actually had its *necessity* established, consequently the Discharge rule is different. The Repetition rule is also different, since strict subordinate proofs involve alternative worlds. The basic rules are in Table 4.

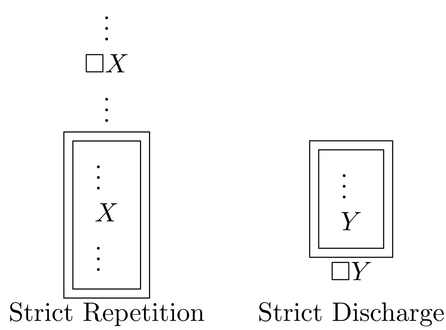


Table 4. Modal Natural Deduction Rules for **K**

An example of a proof using the classical and the modal natural deduction rules can be found in Figure 2. It is a proof of $\Box(X \supset Y) \supset (\Box X \supset \Box Y)$. In it, 1 and 2 are premises; 3 is by Repetition from 1; then a strict subordinate proof is started; 4 and 5 are from 2 and 3 by Strict Repetition; 6 is from 4 and 5 by Modus Ponens; 7 is from 6 by Strict Discharge; 8 is by Discharge; and 9 is by Discharge.

The logic captured by these rules is **K**. Certain other logics can also be treated this way, by suitable additions to the rules. In fact the underlying idea serves us as a lead-in to other proof procedures, starting in the next section. I'll give rules for some, but not

1. $\Box(X \supset Y)$
2. $\Box X$
3. $\Box(X \supset Y)$
4. X
5. $X \supset Y$
6. Y
7. $\Box Y$
8. $\Box X \supset \Box Y$
9. $\Box(X \supset Y) \supset (\Box X \supset \Box Y)$

Figure 2. Modal Natural Deduction Proof

all of the logics from Table 1—treatments of other logics can be found in [10, 17, 55].

If S is a set of formulas, I define a set $S^\sharp$ in a logic-dependent way, in Table 5. The motivation is, if all members of S are true at a possible world of an **L** model, all members of $S^\sharp$ will be true at any alternative world, for **L** being any of the logics listed in Table 5.

Logic	$S^\sharp$
K, **T**, **D**	$\{X \mid \Box X \in S\}$
K4, **S4**, **KD4**	$\{X \mid \Box X \in S\} \cup \{\Box X \mid \Box X \in S\}$
KB, **B**, **S5**	$\{X \mid \Box X \in S\} \cup \{\Box X \mid \Box X \in S\} \cup \{\neg\Box\neg X \mid X \in S\}$

Table 5. Definition of $S^\sharp$

For each of the logics covered in Table 5, the Strict Repetition rule of Table 4 should be replaced by the following. Note that the new rule for **K** coincides with the one stated in Table 4.

Strict Repetition Rule If S is the set of formulas in a proof, that are above a strict subordinate proof, and at the same level as it, any member of $S^\sharp$ can be entered into the strict subordinate proof.

For **K**, **K4**, and **KB**, the rules given so far are complete. For **T**, **B**, **S4**, and **S5**, we add a rule allowing us to infer X from $\Box X$. and for **D** and **KD4**, we instead add a rule allowing us to infer $\neg\Box\neg X$ from $\Box X$.

Figure 3 displays a proof using the rules for **K4**. In it, 1 and 2 are premises; 3 is from 1 by Repetition; 4 is from 2 and 5 is from 3 by Strict Repetition; 6 is from 4 and 7 is from 5 by Strict Repetition; 8 is from 6 and 7 by Modus Ponens; 9 is from 8 by Strict Discharge, as is 10 from 9; 11 is by Discharge, as is 12.

Completeness is easy to show. Begin by giving natural deduction proofs of all (appropriate) axioms. Modus Ponens is one of the rules. And it is easy to show that theorems are closed under a Necessitation Rule (carry out a proof of X inside a strict box, and conclude $\Box X$ outside it). Then natural deduction completeness follows from axiomatic completeness. Soundness is a bit more work; see [19] for details.

1. $\Box(X \supset Y)$
2. $\Box X$
3. $\Box(X \supset Y)$
4. $\Box X$
5. $\Box(X \supset Y)$
6. X
7. $X \supset Y$
8. Y
9. $\Box Y$
10. $\Box\Box Y$
11. $\Box X \supset \Box\Box Y$
12. $\Box(X \supset Y) \supset (\Box X \supset \Box\Box Y)$

Figure 3. **K4** Natural Deduction Proof

While I have been discussing proofs, these natural deduction systems encompass derivations as well. A *local* premise can be added to a derivation provided it is not added inside a strict subderivation. A *global* premise can be added at any point, even inside a strict subderivation. Then soundness and strong completeness can be shown for the logics of Table 5.

5 SEMANTIC TABLEAUS

Both axiom systems and natural deduction are *forward reasoning* systems. One starts with axioms and rules, and finishes with the desired theorem. Such systems, while elegant, are often difficult for proof discovery, and are not good candidates for automation. Various *backward reasoning* systems have been invented. These begin with the desired result and work backward from there to create a proof. For classical logic, *resolution* is such a system—it was designed for machine implementation, and over the years has been the basis for very efficient classical theorem provers, [22, 43]. However, resolution does not tend to adapt well to non-classical logics (though see [20]). Semantic tableaus, or *tableaus* for short, were introduced in [5], and took on their current form independently in [44] and [57]. They too have also had successful computer implementations, and have turned out to be more flexible than resolution in adapting to a rich variety of logics. A very thorough presentation of tableaus can be found in [11]. This section presents tableau systems for several propositional modal logics, but naturally, I'll begin classically.

5.1 *A Classical Tableau System*

Tableaus can be developed using signed or unsigned formulas. I'll present a signed version, and briefly discuss an unsigned one afterward. A *signed* formula is simply $T\,X$ or $F\,X$, where X is a formula. Intuitively, these signed formulas assert that X is true or false respectively, in some context. One begins a proof search with $F\,X$, and

attempts to produce a contradiction, thus showing that X cannot be false under any circumstances. Tableaus take the form of trees, customarily written with the root at the top, and branching downward. Intuitively, each branch represents one 'case.' There are rules for "growing" trees—one for each connective and sign. For axiom systems it was convenient to have a small number of connectives, with others defined from them. There is no corresponding advantage for tableaus, so I'll take $\neg$, $\wedge$, $\vee$, and $\supset$ as primitive from now on, and also $\top$ (truth constant) as well as $\bot$ (falsehood constant). However, $\equiv$ is still best thought of as a defined connective.

'Tree growing' rules involving negation are straightforward, and are given in Table 6. For each, if the signed formula above the line occurs (anywhere) on a tableau branch, the signed formula below can be added to the branch end. Rules for the binary connectives come in groups, and I'll make use of Smullyan's unifying notation here [57]. Table 7 defines what are called *alpha* and *beta* signed formulas and for each, two components. Using this, the binary connective rules are summarized in Table 8. These rules say: if an *alpha* formula occurs on a branch, its two components can be added successively to the branch end; if a *beta* formula occurs, the branch can be split, with one component added to each of the new branch ends.

$$\frac{T\,\neg X}{F\,X} \qquad \frac{F\,\neg X}{T\,X}$$

Table 6. Negation Rules

α	α_1	α_2
$T\,X \wedge Y$	$T\,X$	$T\,Y$
$F\,X \vee Y$	$F\,X$	$F\,Y$
$F\,X \supset Y$	$T\,X$	$F\,Y$

β	β_1	β_2
$T\,X \vee Y$	$T\,X$	$T\,Y$
$F\,X \wedge Y$	$F\,X$	$F\,Y$
$T\,X \supset Y$	$F\,X$	$T\,Y$

Table 7. Alpha and Beta Formulas

$$\frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad \frac{\beta}{\beta_1 \mid \beta_2}$$

Table 8. Alpha and Beta Rules

Figure 4 contains an example of a tree, constructed by starting with the signed formula $F\,\neg(X \wedge Y) \supset (\neg X \vee \neg Y)$. In it, 2 and 3 are from 1 by α; 4 is from 2 by negation; 5 and 6 are from 4 by β, 7 and 8 are from 3 by α, just as 9 and 10 are, on the right branch; 11 is from 7 by negation; 12 is from 10 by negation. Not every applicable rule has been used—neither 8 nor 9 has had a negation rule applied to it.

A tableau branch is called *closed* if it contains $T\,Z$ and $F\,Z$ for some formula Z, or if it contains $F\,\top$, or $T\,\bot$. A tableau is closed if every branch is closed. Intuitively, a closed branch represents an impossible situation, and a closed tableau tells us that every

$$
\begin{array}{ll}
F\,\neg(X \wedge Y) \supset (\neg X \vee \neg Y) & 1. \\
T\,\neg(X \wedge Y) & 2. \\
F\,\neg X \vee \neg Y & 3. \\
F\,X \wedge Y & 4.
\end{array}
$$

$$
\begin{array}{ll}
F\,X & 5. \\
F\,\neg X & 7. \\
F\,\neg Y & 8. \\
T\,X & 11.
\end{array}
\qquad\qquad
\begin{array}{ll}
F\,Y & 6. \\
F\,\neg X & 9. \\
F\,\neg Y & 10. \\
T\,Y & 12.
\end{array}
$$

Figure 4. Propositional Tableau Example

situation is impossible. The tableau of Figure 4 is closed, because of 5 and 11, and 6 and 12.

A *tableau proof* of X is a closed tableau beginning with $F\,X$. Thus Figure 4 constitutes a tableau proof of $\neg(X \wedge Y) \supset (\neg X \vee \neg Y)$. It can be shown that exactly the classical tautologies have tableau proofs in this system, but I'll postpone any discussion of soundness and completeness until modal rules have been introduced. A tableau version of consequence—deduction from premises—is easy. One says X follows from a set S of formulas if there is a closed tableau starting with $F\,X$, allowing the additional rule that for any $Z \in S$, $T\,Z$ can be added to the end of any branch.

Finally, I have used signed formulas, but one could just as well work with an unsigned version. Instead of $F\,X$ use $\neg X$, and instead of $T\,X$, just use X. I'll leave a full formulation to you (or see [22, 57]). The use of signs brings some additional power, however. It is easier to establish a connection with the Gentzen sequent calculus, as we will do in Section 7. There is a simple signed tableau system for intuitionistic logic, something that is not possible without signs. And one can add extra signs to create proof systems for many-valued logics.

5.2 Destructive Modal Tableaus

Modal tableaus come in more than one version. Some logics have *destructive* tableau systems, [19, 33]. These will be presented in this section—a different approach is given in Section 6. The terminology comes from the fact that some destructive tableau rule applications lose information. Destructive tableau proofs tend to be more useful metatheoretically than other kinds of tableaus—for example, one can devise a simple proof of interpolation theorems using such tableau systems.

To continue the uniform treatment begun with the *alpha*/*beta* grouping, two new categories, *nu* and *pi*, and their components are introduced in Table 9, to take care of the modal operators—both $\Box$ and $\Diamond$ are taken as primitive now.

In Table 5 I gave a definition of $S^\sharp$ for several modal logics. As it happens, logics whose semantics involve symmetry don't have simple (or any) destructive tableau systems, so these must be dropped. And I'm now allowing more connectives and modal operators as primitive than before. So a definition appropriate for this section is given in Table 10—

ν	ν_0
$T\,\Box X$	$T\,X$
$F\,\Diamond X$	$F\,X$

π	π_0
$T\,\Diamond X$	$T\,X$
$F\,\Box X$	$F\,X$

Table 9. Nu and Pi Formulas

connections with the earlier version should be clear. A few observations about this definition. First, for all six of the logics we have monotonicity: $S_1 \subseteq S_2$ implies $S_1^\sharp \subseteq S_2^\sharp$. And second, for the **K4**, **S4**, **D4** group we have $S^\sharp \subseteq S^{\sharp\sharp}$. Both of these are easily checked, and both play a role in later soundness and completeness proofs.

Logic	$S^\sharp$
$\mathbf{K}, \mathbf{T}, \mathbf{D}$	$\{\nu_0 \mid \nu \in S\}$
$\mathbf{K4}, \mathbf{S4}, \mathbf{D4}$	$\{\nu_0, \nu \mid \nu \in S\}$

Table 10. Revised Definition of $S^\sharp$

Destructive tableau rules for the logics of Table 10 are as follows. First, all the classical rules of the previous section continue to apply. And in addition there are the rules given in Table 11. These rules require some comment. The second, from ν to get ν_0, is of the same general kind as earlier tableau rules: a branch containing ν can have ν_0 added to the end. The third is slightly different since it is premiseless: at any point on a tableau branch we can add $T\,\Diamond\top$. The first, however, is of a very different nature. Let us call it the π *rule*, though technically what is displayed is actually several rules, depending on the definition of $S^\sharp$. The π rule says that, given a branch containing π, and with S as the entire set of (other) signed formulas on it, that branch can be *replaced* with a new one containing the members of $S^\sharp$, and π_0. The π rule is the reason for the terminology *destructive*—application of this rule removes formulas.

For all logics: $\dfrac{S, \pi}{S^\sharp, \pi_0}$ For **T** and **S4**: $\dfrac{\nu}{\nu_0}$ For **D** and **D4**: $\dfrac{}{T\,\Diamond\top}$

Table 11. Destructive Modal Rules

An example of a destructive tableau can be found in Figure 5. It provides a tableau proof, in the **K4** system, of $(\Box\Diamond X \wedge \Box Y) \supset \Box\Diamond(X \wedge Y)$. In it, 2 and 3 are from 1, and 4 and 5 are from 2 by α. Then a destructive π rule applies, with 3 as the π formula. The original branch is replaced by a new one, shown below the line, with 6 from 3, 7 and 8 from 4, and 9 and 10 from 5; formulas 1 and 2 disappear entirely. Another π rule application now happens, with 8 as the π formula, producing the new branch shown below the second line. Item 11 is from 8; 12 and 13 are from 6; 14 and 15 are from 7; 16 and 17 are from 9. Finally β applied to 13 produces 18 and 19, and both branches are closed.

Destructive rules add a level of complexity to tableaus. Tableau rules are non-deterministic—they say what can be done, but the order of rule application is not specified. It can be shown that, for the classical system of Section 5.1, a kind of Church-Rosser

$$
\begin{array}{l}
F\,(\Box\Diamond X \wedge \Box Y) \supset \Box\Diamond(X \wedge Y) \quad 1. \\
T\,\Box\Diamond X \wedge \Box Y \quad 2. \\
F\,\Box\Diamond(X \wedge Y) \quad 3. \\
T\,\Box\Diamond X \quad 4. \\
T\,\Box Y \quad 5. \\
\hline
F\,\Diamond(X \wedge Y) \quad 6. \\
T\,\Box\Diamond X \quad 7. \\
T\,\Diamond X \quad 8. \\
T\,\Box Y \quad 9. \\
T\,Y \quad 10. \\
\hline
T\,X \quad 11. \\
F\,\Diamond(X \wedge Y) \quad 12. \\
F\,X \wedge Y \quad 13. \\
T\,\Box\Diamond X \quad 14. \\
T\,\Diamond X \quad 15. \\
T\,\Box Y \quad 16. \\
T\,Y \quad 17.
\end{array}
$$

$$
\begin{array}{cc}
F\,X \quad 18. & \qquad\qquad F\,Y \quad 19.
\end{array}
$$

Figure 5. **K4** Destructive Tableau Example

property applies. If a formula X is a tautology, any attempt to provide a closed tableau for $F\,X$ will succeed, no matter in what order the rules are applied, provided only that on each branch, every non-atomic formula eventually has a rule applied to it. The π rule changes things. We might have a tableau branch containing, among other things, both $T\,\Diamond X$ and $T\,\Diamond Y$. Either could be used as π in a π-rule application, but when so used, it will cause the deletion of the other formula. It can happen that a proof is obtainable when one choice is made, but not the other—we can choose badly. This means that a systematic proof search must allow backtracking, and so will be inherently more time-consuming than a systematic search in the classical system—see Chapter 3 for a full discussion.

5.3 Soundness and Completeness

A proof of soundness can also serve to motivate the rules of Table 11. We'll say a signed formula is *realized* at a possible world w of a model $\mathcal{M}$ if the formula is $T\,X$ and $\mathcal{M}, w \Vdash X$, or the formula is $F\,X$ and $\mathcal{M}, w \nVdash X$. Let **L** be one of the six logics for which tableau rules have been provided. Call a set S of signed formulas **L**-*satisfiable* if there is some **L** model $\mathcal{M}$, and some possible world, w, of it that realizes all the members of S. Call a tableau branch **L**-satisfiable if the set of signed formulas on it is **L**-satisfiable. And call a tableau **L**-satisfiable if some branch is. The key fact is that satisfiability is an invariant for tableau construction. That is, each tableau rule preserves **L**-satisfiability. Let us call a rule *sound* if it has this satisfiability preserving feature. It is the need to have sound rules that dictates some of the features of the systems we have seen—for

instance, this is why the version of $S^\sharp$ that works for **K4** will not serve for **K**.

PROPOSITION 5. *Suppose $\mathcal{T}$ is an* **L** *tableau that is* **L***-satisfiable. If any* **L** *tableau rule is applied to $\mathcal{T}$ the resulting tableau is still* **L***-satisfiable.*

Proof. Suppose branch θ of $\mathcal{T}$ is **L**-satisfiable, say its members are realized at world w of model $\mathcal{M}$. And say a tableau rule is applied to $\mathcal{T}$. If it is applied on a branch other than θ, the resulting tableau is trivially **L**-satisfiable, so now assume the tableau rule has been applied on θ.

If the applied rule was a β rule, then θ branches, technically it is replaced with two new branches which we'll call θ, β_1 and θ, β_2, using the obvious notation. Since β was realized at w, a check of each case in the definition of β shows that either β_1 is realized at w, or β_2 is. Consequently, either the branch θ, β_1 is satisfied, at w, or the branch θ, β_2 is. Either way the resulting tableau is **L**-satisfiable. The argument if the rule application was an α or a negation rule is even simpler, and is omitted.

Now suppose the applied rule was the π rule from Table 11. The key thing we need is this. For each of the logics under consideration, if members of S are realized at world w of an **L** model $\mathcal{M}$, and if w' is any world of $\mathcal{M}$ that is accessible from w, then members of $S^\sharp$ are realized at w'. Verification of this is left to you. Now, suppose θ consists of the members of S, and the signed formula π, and w realizes all the signed formulas on θ. Since π is realized at w, there must be an alternate world w' at which π_0 is realized. As we just noted, at w' all members of $S^\sharp$ are realized. Then in the resulting tableau there is still a satisfiable branch, though its members are realized at a different world than the one realizing the members of the original branch.

The other rules from Table 11 are straightforward. ❑

PROPOSITION 6. *If X has a proof using the tableau rules for* **L***, then X is* **L** *valid.*

Proof. I'll show the contrapositive. Suppose X is not **L** valid; so there is some world of some **L**-model at which X is false, Then $\{F\,X\}$ is **L**-satisfiable. Any tableau proof of X must start with the tree with only $F\,X$, at its root. This is an **L**-satisfiable tableau, so Proposition 5 says only **L**-satisfiable tableaus will be produced. An **L**-satisfiable tableau cannot be closed. Hence X can have no **L** tableau proof. ❑

Next we turn to completeness. A common way of showing completeness for tableau systems involves devising a systematic way of applying tableau rules. Such a systematic approach is presented for classical logic in [57], for instance. While such a method has utility when computer implementations are involved, it is often hard work. Fortunately the method used to show completeness for axiom systems in Section 2.2 can also be applied, and is much simpler.

First we need a small generalization of the notion of tableau. So far we have started tableau constructions with a single signed formula. From now on, if S is a finite set of signed formulas, a tableau for S will be any tableau starting with a single branch containing the members of S, and continuing using the usual tableau rules. Then, a tableau proof of a formula X is a closed tableau for the set $\{F\,X\}$.

Let **L** be one of the logics for which tableau rules have been provided in Table 11. Call a set S of *signed* formulas **L**-*inconsistent* if there is a closed **L** tableau for some finite subset of S, and call S **L**-*consistent* if it is not **L**-inconsistent. Clearly this notion of

L-consistency is of finite character, and so the Lindenbaum construction applies, see (1). Every **L**-consistent set of signed formulas can be extended to a maximal **L**-consistent set.

We construct a model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{V} \rangle$ much like we did in Section 2.2. $\mathcal{G}$ is the collection of all maximal **L**-consistent sets of signed formulas. For $w_1, w_2 \in \mathcal{G}$, $w_1 \mathcal{R} w_2$ provided $w_1^\sharp \subseteq w_2$. And finally, $w \in \mathcal{V}(P)$ provided $T\,P \in w$. One cannot, at this point, show an exact counterpart of (2) (though it is, in fact, true). But one can show the following, involving an implication instead of an equivalence. For every *signed* formula $\mathcal{X}$ and possible world $w \in \mathcal{G}$

$$\mathcal{X} \in w \Longrightarrow w \text{ realizes } \mathcal{X} \text{ in the model } \mathcal{M} \tag{4}$$

The proof is by induction on the complexity of signed formulas. Since we have several connectives as primitive now, I'll make use of uniform notation. Here are the cases needed to establish (4).

Suppose P is atomic. If $T\,P \in w$ then $\mathcal{M}, w \Vdash P$ by definition of $\mathcal{V}$, so $T\,P$ is realized at w. Likewise if $F\,P \in w$, since w is **L**-consistent, $T\,P \notin w$, and so $\mathcal{M}, w \nVdash P$, and again $F\,P$ is realized at w.

The negation cases are straightforward, and are omitted.

Suppose we have a β signed formula, $\beta \in w$, and (4) and is known for β_1 and β_2. Since w is **L**-consistent, it follows from the tableau rules that one of $w \cup \{\beta_1\}$ or $w \cup \{\beta_2\}$ is **L**-consistent. Then it follows by maximality of w that either $\beta_1 \in w$ or $\beta_2 \in w$. By the induction hypothesis, either β_1 or β_2 is realized at world w of $\mathcal{M}$. And an examination of the cases in the definition of β formulas shows this is enough for β to be realized at w as well.

The α case is similar, and is omitted.

Suppose we have a ν formula, $\nu \in w$, and (4) is known for ν_0. Let w' be any member of $\mathcal{G}$ with $w \mathcal{R} w'$. For each choice of **L**, $\nu_0 \in w^\sharp$ and since $w^\sharp \subseteq w'$ by definition of $\mathcal{R}$, $\nu_0 \in w'$. By the induction hypothesis, w' realizes ν_0. A check of cases in the definition of ν shows that, since w' was arbitrary, ν is realized at w.

Finally, suppose we have a π formula, $\pi \in w$, and (4) is known for π_0. Using the π rule from Table 11, it follows that $w^\sharp \cup \{\pi_0\}$ is consistent. Let w' be a maximal **L**-consistent extension of this. Then $w' \in \mathcal{G}$ and since $\pi_0 \in w'$, the induction hypothesis gives us that π_0 is realized at w'. Finally, since $w \mathcal{R} w'$, it follows, for each case in the definition of π, that π is realized at w.

Now that (4) has been established, putting the final pieces together is easy. For **L** being any of the six logics from Tables 10 and 11, one can easily check that the construction described above produces a model $\mathcal{M}$ that is, in fact, an **L**-model. Here is part of such a verification. For any of the six logics being treated, $S_1 \subseteq S_2$ implies $S_1^\sharp \subseteq S_2^\sharp$. For **L** being one of **K4**, **S4**, or **D4**, $S^\sharp \subseteq S^{\sharp\sharp}$. Now, suppose $w_1 \mathcal{R} w_2$ and $w_2 \mathcal{R} w_3$. Then $w_1^\sharp \subseteq w_2$ and $w_2^\sharp \subseteq w_3$, so $w_1^{\sharp\sharp} \subseteq w_2^\sharp \subseteq w_3$. So if **L** is one of **K4**, **S4**, or **D4**, $w_1^\sharp \subseteq w_3$, and hence $w_1 \mathcal{R} w_3$. Thus for these three logics, the model is transitive. I'll leave the other conditions to you.

Now, if X is not provable by **L**-tableaus, $\{F\,X\}$ is **L**-consistent. Extend it to a maximal **L**-consistent set w, which will be a world of the **L** model constructed above. It is a world at which X is false, by (4). Thus X is not **L**-valid. This establishes the following.

PROPOSITION 7. *For* **L** *being one of* **K**, **T**, **D**, **K4**, **S4**, **D4**, *the* **L** *tableau rules are complete.*

5.4 The Logic **GL**

In Section 2.3 we saw that, while a straight canonical model argument was not able to prove completeness for **GL**, still a completeness argument could be given. A variation of that argument works for an appropriate destructive tableau system for **GL** as well. I'll sketch this here.

First of all, what are the **GL** destructive tableau rules? Use the definition of $S^\sharp$ for **K4**, from Table 10. In addition, the π-rule itself needs modification. Following [57], define the *conjugate* of a signed formula as follows: $\overline{T\,X} = F\,X$ and $\overline{F\,X} = T\,X$. Thus conjugation amounts to switching the sign. Now, here is the curious but appropriate π rule for **GL**, [8, 19].

$$\frac{S, \pi}{S^\sharp, \pi_0, \overline{\pi}}$$

An example of a proof in this system appears in Figure 6, of the Löb formula $\Box(\Box P \supset P) \supset \Box P$. In it, 2 and 3 are from 1 by α. Then a π-rule application is made, with 3 as the π formula. This replaces the original branch with a new one, shown below the line in the Figure. Formulas 4 and 5 are from 3 and 6 and 7 are from 2. Then 8 and 9 are from 7 by β, and both branches are closed.

$$\begin{array}{ll}
F\,\Box(\Box P \supset P) \supset \Box P & 1. \\
T\,\Box(\Box P \supset P) & 2. \\
F\,\Box P & 3. \\
\hline
F\,P & 4. \\
T\,\Box P & 5. \\
T\,\Box(\Box P \supset P) & 6. \\
T\,\Box P \supset P & 7.
\end{array}$$

$$\begin{array}{lcl}
F\,\Box P \quad 8. & \qquad & T\,P \quad 9.
\end{array}$$

Figure 6. **GL** Destructive Tableau Example

Soundness is shown by the same method as earlier, Proposition 5, and the argument now is just like before, except for the new π rule. So, I'll just show the following: if $S \cup \{\pi\}$ is $\mathbf{GL}^w$-satisfiable, so is $S^\sharp \cup \{\pi_0, \overline{\pi}\}$. Well, suppose $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{V} \rangle$ is a $\mathbf{GL}^w$ model in which the members of $S \cup \{\pi\}$ are realized at possible world w_1, but there is no world at which the members of $S^\sharp \cup \{\pi_0, \overline{\pi}\}$ are realized. Since w_1 realizes π there must be a world, w_2, with $w_1 \mathcal{R} w_2$, with w_2 realizing π_0. And since w_1 realizes the members of S, it is easy to see that w_2 must realize the members of $S^\sharp$. Since no world in $\mathcal{M}$ realizes the members of $S^\sharp \cup \{\pi_0, \overline{\pi}\}$, it must be that w_2 cannot realize $\overline{\pi}$, and hence must realize π—that is, w_2 realizes all members of $S^\sharp \cup \{\pi_0, \pi\}$. But now, since π is realized at w_2, there must be a world w_3 with $w_2 \mathcal{R} w_3$ such that π_0 is realized at w_3. Since $S^\sharp$ is realized at w_2 then $S^{\sharp\sharp}$ is realized at w_3, but for the **K4** definition, $S^\sharp \subseteq S^{\sharp\sharp}$, so all of $S^\sharp$ is realized at w_3. Then, as at w_2, we must have that π is realized at w_3. That is, at w_3, just as at w_2, we have all the members of $S^\sharp \cup \{\pi_0, \pi\}$ realized. This pattern repeats—there must be an accessible world w_4 realizing this set, and so on. But

this contradicts the non-existence of infinite chains in $\mathbf{GL}^w$ models. Thus if $S \cup \{\pi\}$ is $\mathbf{GL}^w$-satisfiable, so is $S^\sharp \cup \{\pi_0, \overline{\pi}\}$, and soundness follows.

For completeness, a modification of the proof in Section 2.3 will work. Let Z be a formula with no **GL** tableau proof—I'll construct a $\mathbf{GL}^f$ counter-model. This time, let $\mathbf{sub}(Z)$ be the set of all signed subformulas of Z—all $T\,X$ and $F\,X$ with X a subformula of Z. Define consistency the way we did in Section 5.3—a set of signed formulas is **GL**-consistent if no **GL** tableau for a finite subset closes. Now, construct a model $\mathcal{M}_Z = \langle \mathcal{G}_Z, \mathcal{R}_Z, \mathcal{V}_Z \rangle$ as follows. $\mathcal{G}_Z$ is the set of all maximally consistent subsets of $\mathbf{sub}(Z)$. A consistent subset of $\mathbf{sub}(Z)$ extends to a maximally consistent subset. Define $w \in \mathcal{V}_Z(P)$ if $T\,P \in w$. Set $w_1 \mathcal{R}_0 w_2$ provided $w_1^\sharp \subseteq w_2$. Then define $w_1 \mathcal{R}_Z w_2$ if $w_1 \mathcal{R}_0 w_2$ but not $w_2 \mathcal{R}_0 w_1$. We now have a model $\mathcal{M}_Z$. Just as in Section 2.3, it is finite, irreflexive, and transitive. Finally, a variant of (3) holds for it. I state it as (5) and leave its proof to you. With it, completeness (but not strong completeness) follows in the usual way.

$$\text{For a signed formula } \mathcal{X} \in \mathbf{sub}(Z),\ \mathcal{X} \in w \text{ implies } w \text{ realizes } \mathcal{X} \tag{5}$$

5.5 Tableau Remarks

Unlike the proof procedures examined earlier in this chapter, tableau systems obey a *subformula principle*—all formulas occurring in a proof are subformulas of the formula being proved. Often this is expressed by saying tableaus are *analytic*. The modus ponens rule of axiom systems and of natural deduction systems is the reason they do not obey a subformula principle. Analyticity makes the finding of proofs a simpler thing and accounts for why tableau systems have frequently been automated while natural deduction and axiom systems have rarely been. Indeed, proofs of decidability for logics having tableau systems can often be based on analyticity.

There is an important non-analytic rule that is sometimes added to tableau systems, the *cut* rule. It says, at any point in a tableau construction we can split the end of a branch, labeling the two new branch nodes with $T\,X$ and $F\,X$, for an *arbitrary* formula X. Since X can be any formula, obviously analyticity is violated. There is a more restricted version of the rule, in which X is required to be a subformula of the formula being proved—this is called *analytic cut.*

Why consider an unrestricted cut rule? Historically, it was introduced by Gentzen [29] in the closely related context of the *sequent calculus* (see Section 7). Gentzen wanted to constructively establish that tableau (sequent calculi) and axiom systems were equivalent for both classical and intuitionistic logic. The presence of a cut rule makes it easy to show this—cut roughly corresponds to modus ponens. Then Gentzen gave a complicated constructive argument that showed any application of a cut rule in a proof could be eliminated. This fact, and its constructive proof, have been very influential, with important consequences, but it is not appropriate to go into this here. Suffice it to say that a cut rule can be added to any of the tableau systems of Section 5.2 without changing the class of provable formulas—that is, cut elimination can be proved for these systems, with proofs going back to [47, 48].

Proofs using cut, at least classically, can be significantly shorter than cut-free proofs. Cut elimination for classical first-order logic can introduce a non-elementary blow-up in proof depth. The corresponding situation for modal logics seems not to have been much studied, but is probably similar. There has been recent work on designing proof systems

making proofs of cut elimination easier to establish. These go under the name *display logics*. See [4, 58].

Cut elimination for our modal systems can be shown constructively by extending Gentzen's argument, but the arguments are fussy. A non-constructive proof is quite simple, however. The cut rule is easily seen to be a sound rule—it preserves **L**-satisfiability, for each **L** we have considered. Then the soundness proof of Section 5.3 extends—if X has an **L**-tableau proof, allowing the cut rule, X must be **L** valid. But by our completeness result, Proposition 7, if X is **L** valid it must have an **L**-tableau proof without cut. Consequently, if X is provable using **L**-tableaus plus cut, X is provable using **L**-tableaus without cut.

A rule that can be added to a proof procedure without changing the class of theorems is called an *admissible* rule. What we have just shown is that cut is an admissible rule. Incidentally, now that we know this, it is easy to see that the implication (4) is actually an equivalence. This follows since, using cut, for each formula X, either $T\,X$ or $F\,X$ must be in any maximal **L**-consistent set, and closed tableaus using cut can be replaced by closed tableaus not using cut.

Why consider analytic cut? For one thing, it can shorten proofs, and does not violate the subformula principle, so proof search procedures can incorporate it in a reasonable way. It has sometimes been included in tableau implementations for this reason. The reader cannot fail to have noticed that while nine representative modal logics were introduced in Table 1, tableau systems were given for only six of them. Tableau systems for the other three are missing, though if analytic cut is allowed, destructive tableau systems can be created, [33]. In this respect our representative normal modal logics are actually representative—more logics have axiom systems than have cut-free destructive tableau systems. Since tableau systems are useful for automation, a number of attempts have been made to augment tableaus with additional machinery so that more logics can be covered. We will see more of this starting in Section 6.

Finally there is the matter of deduction, and the possibility of a strong completeness theorem. For the logics of Section 5.2, this is an easy matter. If a formula X is a *global* premise, one is allowed to add $T\,X$ to any open tableau branch at any point. If X is a *local* premise, one can add it to any open tableau branch provided the π-rule in Table 11 has not yet been applied on that branch. Then our soundness and completeness arguments do, in fact, extend to prove strong soundness and completeness—I omit the proof. It should be noted that there are logics, **GL** is an example, that have sound and complete tableau systems, but which do not extend to allow these additional premise-adding rules. (Otherwise one could establish compactness for **GL**, and there is a simple example in [19] showing that local compactness fails.) Things must be carefully checked.

6 PREFIXED TABLEAUS

In a destructive modal tableau an application of a π rule corresponds to a move to an alternative world—this appears explicitly in the argument for the soundness of the π rule. But the π rule loses information. If we are attempting to apply such a rule in a logic whose semantics involves symmetry, we can expect problems. With symmetry we can leave a world and return to it, so to speak. But in a destructive tableau, having lost information, there is no mechanism to regain it (except, sometimes, analytic cut). In order to get around this problem various mechanisms have been introduced to retain

information about other worlds during a proof—for instance in [37] a method of *semantic diagrams* is presented, in which multiple-world information is retained by the use of multiple boxes. *Prefixed tableaus* provide additional machinery using a device that is especially syntactic in nature. These tableau systems were introduced in [18, 19], but took on their present modular form in [46, 33]. They can also be seen as a particular kind of *labeled deductive system*, see [27].

6.1 A Prefixed System for **K**

A *prefix* is a finite sequence of positive integers, which I will write using a dot as separator, for example, 1.2.3.2.3. The informal idea is that a prefix names a possible world, with 1.2.3.2.1, 1.2.3.2.2, 1.2.3.2.3, and so on, all naming worlds accessible from 1.2.3.2. A *prefixed (signed) formula* is $\sigma\,T\,X$ or $\sigma\,F\,X$, where σ is a prefix and X is a formula. The informal idea is that a prefixed formula asserts the underlying formula is true/false at the world named by the prefix.

A prefixed tableau proof of X begins with $1\,F\,X$, informally asserting that X is false in some world, named by 1. It continues using branch extension rules to be given in a moment. The goal is to produce a closed tableau, where now a branch is closed if it contains $\sigma\,T\,X$ and $\sigma\,F\,X$ for some formula X (note that the prefix is the same in both cases). A branch is also closed if it contains $\sigma\,T\,\bot$ or $\sigma\,F\,\top$. And a tableau is closed if each branch is closed, as usual.

The branch extension rules for the propositional connectives are as before, except that prefixes are carried along. They are given in Table 12.

$$\frac{\sigma\,T\,\neg X}{\sigma\,F\,X} \qquad \frac{\sigma\,F\,\neg X}{\sigma\,T\,X} \qquad \frac{\sigma\,\alpha}{\begin{array}{c}\sigma\,\alpha_1\\ \sigma\,\alpha_2\end{array}} \qquad \frac{\sigma\,\beta}{\sigma\,\beta_1 \quad \mid \quad \sigma\,\beta_2}$$

Table 12. Prefixed Classical Rules

The modal rules for **K** are given in Table 13. A prefix has been *used* on a branch if it already occurs on the tableau branch. It is *new* if it does not occur. The intuition should be fairly clear. If a π formula is true at a world named by prefix σ, then π_0 must be true at an alternative world. We want to pick a name for that world, a prefix. Since the world is accessible from the world named by σ, we want a prefix that extends σ by one number, and otherwise it should be uncommitted, hence the newness requirement. The ν rule is similarly motivated.

$$\frac{\sigma\,\nu}{\begin{array}{c}\sigma.n\,\nu_0\\ \text{for } \sigma.n \text{ used}\end{array}} \qquad \frac{\sigma\,\pi}{\begin{array}{c}\sigma.n\,\pi_0\\ \text{for } \sigma.n \text{ new}\end{array}}$$

Table 13. Modal Tableau Rules for **K**

Figure 7 contains an example of a proof in this **K** tableau system, of $(\Diamond\Box P \wedge \Box\Diamond Q) \supset \Diamond\Diamond(P \wedge Q)$. In it, 2 and 3 are from 1, and 4 and 5 are from 2 by α, 6 is from 4 by π, 7

is from 5 by ν, 8 is from 7 by π, 9 is from 3 and 10 is from 9 by ν, 11 and 12 are from 10 by β, and 13 is from 6 by ν. Closure is by 11, 13 and 8, 12.

$$\begin{array}{lll}
1 & F\,(\Diamond\Box P \wedge \Box\Diamond Q) \supset \Diamond\Diamond(P \wedge Q) & 1. \\
1 & T\,\Diamond\Box P \wedge \Box\Diamond Q & 2. \\
1 & F\,\Diamond\Diamond(P \wedge Q) & 3. \\
1 & T\,\Diamond\Box P & 4. \\
1 & T\,\Box\Diamond Q & 5. \\
1.1 & T\,\Box P & 6. \\
1.1 & T\,\Diamond Q & 7. \\
1.1.1 & T\,Q & 8. \\
1.1 & F\,\Diamond(P \wedge Q) & 9. \\
1.1.1 & F\,P \wedge Q & 10.
\end{array}$$

$$\begin{array}{lll|lll}
1.1.1 & F\,P & 11. & 1.1.1 & F\,Q & 12. \\
1.1.1 & T\,P & 13. & & &
\end{array}$$

Figure 7. Prefixed **K** Tableau

Prefixed tableaus can be used for derivations as well as for proofs, in quite a simple way. To use X as a *global* premise, one may add $\sigma\,T\,X$ to the end of any open branch, for any prefix σ that appears on the branch. To use X as a *local* premise, one may add $1\,T\,X$ only.

6.2 Soundness and Completeness

So far I have only given prefixed rules for **K**. But rules for other logics do not change the basic ideas very much, so I'll prove soundness and completeness now, while things are at their simplest.

Soundness is by the usual tableau argument—see Section 5.3 for details in the destructive tableau setting. For prefixed formulas, I'll say a set S of prefixed, signed formulas is *satisfiable* (properly speaking, **K** satisfiable) if there is a model $\mathcal{M}$, and a mapping $\mathcal{N}$ from the prefixes in S to possible worlds in $\mathcal{M}$, such that if $\sigma\,\mathcal{X} \in S$ then $\mathcal{X}$ is realized at $\mathcal{N}(\sigma)$ in $\mathcal{M}$, where $\mathcal{X}$ is a signed formula. As before, a tableau branch is satisfiable if the set of prefixed formulas on it is satisfiable, and a tableau is satisfiable if some branch is. I'll leave it to you to establish that each tableau rule converts a satisfiable tableau into another satisfiable tableau. And trivially, a closed tableau cannot be satisfiable. Now soundness follows exactly as in Section 5.3.

Modal operators have strong similarities to quantifiers, and that observation plays a role now. For starters, the Lindenbaum construction of (1) needs to be 'Henkinized.' We'll say a set S of prefixed signed formulas is **K**-*consistent* if no **K**-tableau for a finite part of S is closed; S is π-*complete* provided, if $\sigma\,\pi \in S$ then for some integer k, $\sigma.k\,\pi_0 \in S$; and S *omits infinitely many integers* if the set of integers that do not appear in prefixes in S is infinite. It is a fact that every **K**-consistent set S of prefixed sentences that omits infinitely many integers can be extended to a set that is maximally **K**-consistent and

π-complete. This can be done via the following Henkin-style modification of the earlier construction, (1).

Lindenbaum-Henkin Construction Suppose S is a **K**-consistent set of prefixed sentences that omits infinitely many integers. Enumerate the (countably many) prefixed signed formulas of the language, $\sigma_1\,\mathcal{X}_1$, $\sigma_2\,\mathcal{X}_2$, ..., and define the following sequence of sets.

$$S_1 = S$$

$$S_{n+1} = \begin{cases} S_n \cup \{\sigma_n\,\mathcal{X}_n\} & \text{if } \mathbf{K}\text{-consistent and } \mathcal{X}_n \text{ is not } \pi \\ S_n \cup \{\sigma_n\,\pi, \sigma_n.k\,\pi_0\} & \text{if } \mathbf{K}\text{-consistent, } \mathcal{X}_n \text{ is } \pi\text{, and } \sigma_n.k \text{ is new} \\ S_n & \text{otherwise} \end{cases} \tag{6}$$

In this construction, 'new' means $\sigma_n.k$ does not occur in S_n or in $\pi(=\mathcal{X}_n)$.

It is not hard to see that if S omits infinitely many integers, this will also be the case with each S_n. Also, if $S \cup \{\sigma_n\,\pi\}$ is **K**-consistent, then $S_n \cup \{\sigma_n\,\pi, \sigma_n.k\,\pi_0\}$ will also be **K**-consistent provided $\sigma_n.k$ is new, and if S_n omits infinitely many integers, there will be such a prefix that is new. I'll leave it to you to check that if S is **K**-consistent and omits infinitely many integers, then $\cup_n S_n$ will be maximally **K**-consistent and π-complete.

Suppose X is not provable using the prefixed **K**-tableau rules. Then $\{1\,F\,X\}$ is **K**-consistent, and obviously omits infinitely many integers. Extend it to a maximally **K**-consistent, π-complete set, S, using the construction above. Let $\mathcal{G}$ be the set of prefixes that occur in S. For prefixes σ and τ in $\mathcal{G}$, set $\sigma\mathcal{R}\tau$ if τ is $\sigma.n$ for some integer n. For a propositional letter P, let $\sigma \in \mathcal{V}(P)$ if $\sigma\,T\,P \in S$. This gives us a model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{V}\rangle$. Incidentally, note that the model is constructed from a single maximally consistent set, rather than a family of them, as was the case with destructive tableaus. This is a key difference between the two types of tableaus: prefixed tableau branches keep track of multiple worlds; destructive tableau branches have information about a single world at a time.

Now we need a truth lemma. It says: for every signed formula $\mathcal{X}$, the following is true.

$$\sigma\,\mathcal{X} \in S \Longrightarrow \sigma \text{ realizes } \mathcal{X} \text{ in the model } \mathcal{M} \tag{7}$$

Equation (7) has a straightforward proof, which I'll leave to you. Once we have it completeness is immediate, since $\{1\,F\,X\} \in S$ and so X is false at world 1 of the model $\mathcal{M}$.

The completeness proof just given is simple, but there is another way of proving completeness that provides additional information. One can give an algorithm that systematically expands **K** tableaus. If the algorithm is properly crafted, one can show that either it will produce a proof, or it will terminate with an unclosed tableau. If it terminates, the set of formulas on any open branch can play the role of the set S in the proof above; (7) can be proved for it. In this way we not only get completeness, but a concrete decision procedure as well. Such an algorithm is given in [19, Ch 8, sect 4]. With some of the other logics to be discussed in the next section, termination is not as simple as it is with **K**, and may involve loop-checking. I do not pursue this further here.

T	$\dfrac{\sigma\,\nu}{\sigma\,\nu_0}$	$4r$	$\dfrac{\sigma.n\,\nu}{\sigma\,\nu}$		
4	$\dfrac{\sigma\,\nu}{\sigma.n\,\nu}$	D	$\dfrac{\sigma\,T\,\Box X}{\sigma\,T\,\Diamond X}$	$\dfrac{\sigma\,F\,\Diamond X}{\sigma\,F\,\Box X}$	
B	$\dfrac{\sigma.n\,\nu}{\sigma\,\nu_0}$				

For prefixes σ and $\sigma.n$ already occurring on the tableau branch:

Table 14. Modal Tableau Rules

6.3 Other Modal Logics

Following [46, 33], other standard modal logics can be handled in a modular fashion. First, some additional tableau rules, and their names, are given in Table 14.

Next, some common modal logics can be given prefixed tableau proof systems by adding various combinations of these rules to those for **K**. How to do this for several modal logics is summarized in Table 15. I omit proofs of soundness and completeness—they are straightforward variants of what worked for **K**.

Logic	Special Rules
T	T
K4	4
S4	T, 4
KB	B
B	B, 4
S5	T, 4, $4r$
D	D
D4	D, 4
DB	D, B

Table 15. Prefixed Tableau Systems

Note that, unlike with destructive modal tableaus, there are straightforward prefixed tableau systems for logics involving symmetry in their semantics. In particular, there is a system for **S5**. In this case, and this case only, there is actually a simpler version that will also serve—let us call it the *Simple* **S5** *System*. Instead of taking prefixes to be sequences of positive integers, take them to be *single* positive integers. And replace the modal rules of Tables 13 and 15 with those in Table 16. Essentially, this works because there is an alternate Kripke semantics for **S5** in which the accessibility relation holds between any two worlds.

Figure 8 displays a proof using these Simple **S5** Rules, of $P \supset \Box\Diamond P$.

$$\frac{n\,\nu}{k\,\nu_0}\ \text{for } k \text{ used} \qquad \frac{n\,\pi}{k\,\pi_0}\ \text{for } k \text{ new}$$

Table 16. Simple Tableau Rules just for **S5**

$$\begin{array}{l}
1\,F\,P \supset \Box\Diamond P \quad 1.\\
1\,T\,P \quad 2.\\
1\,F\,\Box\Diamond P \quad 3.\\
2\,F\,\Diamond P \quad 4.\\
1\,F\,P \quad 5.
\end{array}$$

Figure 8. A Prefixed **S5** Tableau Proof

It was shown in [42] that for **K**, **T**, and **S4**, satisfiability is PSpace complete (more generally, for modal logics between **K** and **S4**). It was also shown that for **S5** it drops to NP complete. This is reflected in the existence of the simple tableau system for **S5** given here. On the other hand, for multi-modal logics, which will be considered in Section 9, satisfiability is PSpace complete, [34], even for multi-modal **S5**, and so we should not expect a simple multi-modal version for this logic, unlike in the mono-modal case. A thorough discussion of complexity issues can be found in this volume, in Chapter 3.

7 GENTZEN SYSTEMS

A *sequent* is written $X_1, \ldots, X_n \longrightarrow Y_1, \ldots, Y_k$, where the X_i and Y_j are formulas. It is informally taken to mean the conjunction of the formulas on the left of the arrow has the disjunction of the formulas on the right as a consequence. Either n or k can be 0, with an empty conjunction treated as true, and an empty disjunction as false. A sequent calculus is a specification of rules for deriving sequents from sequents. But how something is *written* does not determine its mathematical properties. What actually is a sequent?

A list of formulas, $X_1, \ldots, X_n$ say, in a sequent can represent at least three different mathematical objects: a *set*, a *multiset*, or a *sequence*. In Gentzen's original treatment, [29], a formula list represented a sequence. Gentzen provided rules permitting, for example, the permutation of members of a list, or the duplication of members. If a formula list is taken to represent a multiset, permutation need not be specified, but formula duplication still must be provided for. If a formula list is taken to represent a set, even duplication rules can be omitted. Rules like permutation, duplication, and a few others, are called *structural* rules. By using a multiset or a sequence, and at the same time omitting various structural rules, a family of *substructural* logics has been created, with linear logic and relevance logic as the best-known representatives, [14, 50, 53]. By restricting the right side of a sequent to have at most one formula, *intuitionistic* logic can be captured. This is not the place to go into what is a very extensive subject—here we are interested in modal logic over a classical logic base. Consequently here lists will be thought of as designating *sets*, which means structural rules are not needed—they are

built into the data structure, so to speak.

7.1 Classical Propositional Sequents

With it understood that lists represent sets, the Gentzen system rules for classical propositional logic will now be given. I use boldface letters, with and without subscripts, to denote formula lists, which represent sets. If I write $\mathbf{S}, X$ I mean the list consisting of the members of $\mathbf{S}$, together with X, and similarly for other obvious notational conventions. A sequent calculus is a forward reasoning system: certain sequents are taken as axioms, and there are rules for deducing a sequent from others. Note that the rules each introduce exactly one connective, and there is one rule for an introduction on the left of the arrow, and one for the right.

Axioms

$$\mathbf{S}_L, X \longrightarrow \mathbf{S}_R, X$$
$$\mathbf{S}_L, \bot \longrightarrow \mathbf{S}_R$$
$$\mathbf{S}_L \longrightarrow \mathbf{S}_R, \top$$

Negation Rules

$$\frac{\mathbf{S}_L, X \longrightarrow \mathbf{S}_R}{\mathbf{S}_L \longrightarrow \mathbf{S}_R, \neg X}$$

$$\frac{\mathbf{S}_L \longrightarrow \mathbf{S}_R, X}{\mathbf{S}_L, \neg X \longrightarrow \mathbf{S}_R}$$

Conjunction Rules

$$\frac{\mathbf{S}_L, X, Y \longrightarrow \mathbf{S}_R}{\mathbf{S}_L, X \wedge Y \longrightarrow \mathbf{S}_R}$$

$$\frac{\mathbf{S}_L \longrightarrow \mathbf{S}_R, X \qquad \mathbf{S}_L \longrightarrow \mathbf{S}_R, Y}{\mathbf{S}_L \longrightarrow \mathbf{S}_R, X \wedge Y}$$

Disjunction Rules

$$\frac{\mathbf{S}_L, X \longrightarrow \mathbf{S}_R \qquad \mathbf{S}_L, Y \longrightarrow \mathbf{S}_R}{\mathbf{S}_L, X \vee Y \longrightarrow \mathbf{S}_R}$$

$$\frac{\mathbf{S}_L \longrightarrow \mathbf{S}_R, X, Y}{\mathbf{S}_L \longrightarrow \mathbf{S}_R, X \vee Y}$$

Implication Rules

$$\frac{\mathbf{S}_L, Y \longrightarrow \mathbf{S}_R \qquad \mathbf{S}_L \longrightarrow X, \mathbf{S}_R}{\mathbf{S}_L, X \supset Y \longrightarrow \mathbf{S}_R}$$

$$\frac{\mathbf{S}_L, X \longrightarrow \mathbf{S}_R, Y}{\mathbf{S}_L \longrightarrow \mathbf{S}_R, X \supset Y}$$

A Gentzen system proof of a sequent is a tree having the sequent at its root—customarily the root is written at the bottom—with axioms at leaves, and with each non-leaf following from its children by one of the rules above. A proof of a formula X is taken to be a proof of the sequent $\longrightarrow X$. Figure 9 shows a proof in this system, of $\neg(X \wedge Y) \supset (\neg X \vee \neg Y)$. I'll leave it to you to supply reasons for the steps.

Soundness for this sequent calculus is easily established. Define a mapping from se-

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{X \longrightarrow \neg Y, X}{\longrightarrow \neg X, \neg Y, X}}{\longrightarrow \neg X \vee \neg Y, X} \qquad \dfrac{\dfrac{Y \longrightarrow \neg X, Y}{\longrightarrow \neg X, \neg Y, Y}}{\longrightarrow \neg X \vee \neg Y, Y}}{\longrightarrow \neg X \vee \neg Y, X \wedge Y}}{\neg(X \wedge Y) \longrightarrow \neg X \vee \neg Y}}{\longrightarrow \neg(X \wedge Y) \supset (\neg X \vee \neg Y)}$$

Figure 9. Gentzen Sequent Proof

quents to formulas as follows.

$$\begin{aligned} [X_1, \ldots, X_n \longrightarrow Y_1, \ldots, Y_k]^f &= [(X_1 \wedge \cdots \wedge X_n) \supset (Y_1 \vee \cdots \vee Y_k)] \\ [X_1, \ldots, X_n \longrightarrow]^f &= [(X_1 \wedge \cdots \wedge X_n) \supset \bot] \\ [\longrightarrow Y_1, \ldots, Y_k]^f &= [\top \supset (Y_1 \vee \cdots \vee Y_k)] \end{aligned} \tag{8}$$

This mapping is in keeping with the remarks at the beginning of the section about the informal meaning of a sequent—conjunctions entailing disjunctions. It is easy to show that if images of the premises of a sequent rule are classically valid formulas, so is the image of the conclusion. Also, images of all sequent axioms are classically valid formulas. So the image of every provable sequent, under this mapping, is valid. If a formula X has a sequent proof, the sequent $\longrightarrow X$ is provable, hence its image is valid. But $[\longrightarrow X]^f$ is $\top \supset X$, and it follows that X is valid.

Gentzen systems predate tableaus by many years. They were introduced as a tool for the analysis of proofs, while tableaus were introduced as a convenient mechanism for proof search. But tableaus were heavily influenced by Gentzen systems—indeed there is a simple correspondence between sequents and sets of signed (unprefixed) formulas. Define a mapping from finite sets of signed formulas to sequents as follows.

$$\{T\,X_1, \ldots, T\,X_n, F\,Y_1, \ldots, F\,Y_k\}^s \text{ is the sequent } X_1, \ldots, X_n \longrightarrow Y_1, \ldots, Y_k \tag{9}$$

Thus the mapping puts T-signed formulas on the left and F-signed formulas on the right. Now, a key fact is the following.

LEMMA 8. *Let S be a finite set of signed formulas. If there is a closed tableau for S using the classical tableau rules of Section 5.1, then the sequent S^s is provable using the sequent rules of this section.*

Proof. Let us say S *closes with depth* d if d is the smallest number such that there is a closed tableau for S with d tableau rule applications. The proof is by induction on d.

If S closes with depth 0, it must contain $T\,X$ and $F\,X$ for some X, or $T\,\bot$, or $F\,\top$. In each case S^s is a sequent calculus axiom.

Now suppose S closes with depth d and the result is known for sets that close with depth less than d. Say that $T\,X \wedge Y \in S$ and in a d-step tableau for S the first rule application is to this signed formula. It follows that there must be a closed tableau for $S \cup \{T\,X, T\,Y\}$ with fewer than d rule applications. By the induction hypothesis, the sequent image of this set must have a sequent calculus proof. Let $\mathbf{S}_L$ be the list

of formulas that occur in S with a sign of T, and let $\mathbf{S}_R$ be the list of formulas in S with a sign of F. Then $[S \cup \{T\,X, T\,Y\}]^s$ is the sequent $\mathbf{S}_L, X, Y \longrightarrow \mathbf{S}_R$, and so this must be provable. Then by one of the two sequent rules for conjunction, the sequent $\mathbf{S}_L, X \wedge Y \longrightarrow \mathbf{S}_R$ is provable, but this is S^s.

The other cases are similar. ❑

With this established, we quickly get the following.

PROPOSITION 9. *The classical propositional sequent calculus is complete.*

Proof. For Proposition 7 a completeness proof for a destructive modal tableau system was given. It is easy to see that a completeness proof for the non-modal part is extractable. Leaving this to you, we have that if X is classically valid, there must be a closed classical tableau for $\{F\,X\}$. By the Lemma above, the sequent $\longrightarrow X$ must then be provable, and hence X has a sequent calculus proof. ❑

It is important to understand what is behind the proof of Lemma 8. In the one induction case given in detail, a tableau and a sequent rule for conjunction were involved. Here they are, side by side.

$$\frac{T\,X \wedge Y}{\begin{array}{c}T\,X\\ T\,Y\end{array}} \qquad\qquad \frac{\mathbf{S}_L, X, Y \longrightarrow \mathbf{S}_R}{\mathbf{S}_L, X \wedge Y \longrightarrow \mathbf{S}_R}$$

Notice that, via the mapping from sets of signed formulas to sequents, these rules correspond in a fairly obvious way, except that each is an upsidedown version of the other. This is the case with every tableau and sequent rule. It is what makes Lemma 8 work. Once this correspondence is understood, it is easy to see that the sequent proof in Figure 9 and the tableau proof in Figure 4 are just presentations, in their respective systems, of the same construction. The correspondence between tableaus and Gentzen system proofs is worked out in detail in [57], where the two systems are developed simultaneously.

7.2 Modal Propositional Sequents

Now that the correspondence between tableau proofs and sequent proofs is clear in the classical case, we have a guiding principle to follow in turning destructive modal tableau rules into modal sequent rules. We want them to be 'upside down' counterparts. If we do it properly we are guaranteed completeness, since we already have tableau completeness proofs. Here is a sequent version for **K** that does this. First, the definition of $S^\sharp$ needs a dual version.

$$S^\sharp = \{X \mid \Box X \in S\} \qquad\qquad S^\flat = \{X \mid \Diamond X \in S\}$$

And now, the additional rules needed for **K** are the following.

$$\frac{\mathbf{S}_L^\sharp, X \longrightarrow \mathbf{S}_R^\flat}{\mathbf{S}_L, \Diamond X \longrightarrow \mathbf{S}_R} \qquad\qquad \frac{\mathbf{S}_L^\sharp \longrightarrow \mathbf{S}_R^\flat, X}{\mathbf{S}_L \longrightarrow \mathbf{S}_R, \Box X}$$

For **K4** the definitions above must be replaced with the following, though the form of the rules stays the same.

$$S^\sharp = \{\Box X, X \mid \Box X \in S\} \qquad\qquad S^\flat = \{\Diamond X, X \mid \Diamond X \in S\}$$

For **S4** we use the system for **K4** together with the following additional rules, the **T** ones.

$$\frac{\mathbf{S}_L, X \longrightarrow \mathbf{S}_R}{\mathbf{S}_L, \Box X \longrightarrow \mathbf{S}_R} \qquad \frac{\mathbf{S}_L \longrightarrow \mathbf{S}_R, X}{\mathbf{S}_L \longrightarrow \mathbf{S}_R, \Diamond X}$$

Clearly, any of the modal logics with a destructive tableau system also has a sequent calculus proof system.

8 HYPERSEQUENTS

Just as tableaus have been fitted out with extra machinery, such as prefixes, sequent calculi have also been enhanced. *Hypersequents* add machinery that makes it possible to provide proof systems for several well-known non-classical logics—see [3]. While they provide a uniform mechanism that can deal with a rich variety of logics, here I will only discuss the hypersequent calculus for **S5**.

8.1 Hypersequents for **S5**

A hypersequent is written $\mathbf{X}_1 \longrightarrow \mathbf{Y}_1 \mid \mathbf{X}_2 \longrightarrow \mathbf{Y}_2 \mid \cdots \mid \mathbf{X}_n \longrightarrow \mathbf{Y}_n$. In this expression each *component*, $\mathbf{X}_i \longrightarrow \mathbf{Y}_i$, is a sequent (so each $\mathbf{X}_i$ and $\mathbf{Y}_i$ is a list of formulas), and the expression itself is a list of sequents. Intuitively a hypersequent should be read disjunctively—one of the sequents is the case—though exact details vary from logic to logic. Various non-classical logics can be captured by imposing special conditions on how lists behave, but for **S5** things are rather simple. As we did in Section 7, a list will be thought of as designating a *set*, and hence permutation and repetition is built in.

A proof of a formula X is a proof of the hypersequent with the single component $\rightarrow X$, where the notion of a proof of a hypersequent is about to be defined. Much as with a Gentzen system, there are axioms, and rules for deducing hypersequents from hypersequents.

Axioms are component-wise versions of those in Section 7.1: a hypersequent $\mathcal{C}_1 \mid \mathcal{C}_2 \mid \cdots \mid \mathcal{C}_n$ is an axiom if some $\mathcal{C}_i$ is an axiom in the sequent sense. The rules for the classical propositional connectives also carry over component-wise. Here are the rules for conjunction, as an example. In them, the $\mathcal{C}_i$ and $\mathcal{D}_i$ are sequents.

$$\frac{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{S}_L, X, Y \longrightarrow \mathbf{S}_R \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{S}_L, X \wedge Y \longrightarrow \mathbf{S}_R \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}$$

$$\frac{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{S}_L \longrightarrow \mathbf{S}_R, X \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k \qquad \mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{S}_L \longrightarrow \mathbf{S}_R, Y \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{S}_L \longrightarrow \mathbf{S}_R, X \wedge Y \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}$$

We want a proof system for **S5**. Since this extends **S4**, besides the classical rules we need the **S4** modal rules, from the end of Section 7.2, in hypersequent form of course. There are four such **S4** rules. First there are the **K4** rules, in which $S^\sharp = \{\Box X, X \mid \Box X \in S\}$ and $S^\flat = \{\Diamond X, X \mid \Diamond X \in S\}$.

$$\frac{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{S}_L^\sharp, X \longrightarrow \mathbf{S}_R^\flat \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{S}_L, \Diamond X \longrightarrow \mathbf{S}_R \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}$$

$$\frac{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{S}_L^\sharp \longrightarrow \mathbf{S}_R^\flat, X \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{S}_L \longrightarrow \mathbf{S}_R, \Box X \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}$$

And here are the hypersequent **T** rules.

$$\frac{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{S}_L, X \longrightarrow \mathbf{S}_R \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{S}_L, \Box X \longrightarrow \mathbf{S}_R \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}$$

$$\frac{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{S}_L \longrightarrow \mathbf{S}_R, X \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{S}_L \longrightarrow \mathbf{S}_R, \Diamond X \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}$$

So far the rules have made no special use of the hypersequent mechanism; they have all been direct counterparts of sequent calculus rules. There is one final rule that turns the system into one for **S5**, and in it the full mechanism comes into play. First some terminology. Call a formula *modal* if it is of the form $\Box X$ or $\Diamond X$. I'll say the pair of sequents $\mathbf{X}_1 \longrightarrow \mathbf{Y}_1$ and $\mathbf{X}_2 \longrightarrow \mathbf{Y}_2$ is a *modal splitting* of the sequent $\mathbf{U} \longrightarrow \mathbf{V}$ if, first, $\mathbf{X}_1 \cup \mathbf{X}_2 = \mathbf{U}$ and $\mathbf{Y}_1 \cup \mathbf{Y}_2 = \mathbf{V}$ and, second, all formulas in $\mathbf{X}_1$ and in $\mathbf{Y}_1$ are modal. (It is not assumed that $\mathbf{X}_1$ and $\mathbf{X}_2$ are disjoint, and similarly for $\mathbf{Y}_1$ and $\mathbf{Y}_2$).

Modal Splitting Rule If the sequents $\mathcal{C}_i^1$ and $\mathcal{C}_i^2$ are a modal splitting of $\mathcal{C}_i$, then:

$$\frac{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_{i-1} \mid \mathcal{C}_i \mid \mathcal{C}_{i+1} \mid \cdots \mid \mathcal{C}_n}{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_{i-1} \mid \mathcal{C}_i^1 \mid \mathcal{C}_i^2 \mid \mathcal{C}_{i+1} \mid \cdots \mid \mathcal{C}_n}$$

8.2 Examples

I'll first give some useful derived rules, and then a hypersequent proof that makes use of them. The first derived rule is *weakening*, which allows the introduction of additional formulas into sequents. It says the following.

Weakening Rule

$$\frac{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X} \longrightarrow \mathbf{Y} \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X}, \mathbf{U} \longrightarrow \mathbf{Y}, \mathbf{V} \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}$$

Sometimes weakening is taken as a basic rule in the sequent calculus, but in our case it is built in indirectly. Note that sequent axioms were not of the form $X \longrightarrow X$, but allowed *side formulas*, $\mathbf{S}_L, X \longrightarrow \mathbf{S}_R, X$, and similarly for the other two axiom forms. All the various rules allow us to carry side formulas along, and also the **K** rules allow for the introduction of additional side formulas. Given this, showing that weakening is a derived rule is an easy induction on proof complexity.

The next derived rules are peculiar to hypersequents—there is no underlying sequent version.

Derived ν Rules

$$\frac{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X}, \Box A \longrightarrow \mathbf{Y} \mid \mathbf{U}, A \longrightarrow \mathbf{V} \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X}, \Box A \longrightarrow \mathbf{Y} \mid \mathbf{U} \longrightarrow \mathbf{V} \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}$$

$$\frac{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X} \longrightarrow \mathbf{Y}, \Diamond A \mid \mathbf{U} \longrightarrow \mathbf{V}, A \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X} \longrightarrow \mathbf{Y}, \Diamond A \mid \mathbf{U} \longrightarrow \mathbf{V} \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}$$

Derived π Rules

$$\frac{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X}, \Diamond A \longrightarrow \mathbf{Y} \mid A \longrightarrow \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X}, \Diamond A \longrightarrow \mathbf{Y} \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}$$

$$\frac{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X} \longrightarrow \mathbf{Y}, \Box A \mid \longrightarrow A \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}{\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X} \longrightarrow \mathbf{Y}, \Box A \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k}$$

Here is a hypersequent derivation showing that the first of the ν rules above is a derived rule. The *Repetition* label refers to the fact that for us lists represent sets, and so repeated sequents can be combined.

$$\begin{array}{cl}
\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X}, \Box A \longrightarrow \mathbf{Y} \mid \mathbf{U}, A \longrightarrow \mathbf{V} \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k & \\ \hline
\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X}, \Box A \longrightarrow \mathbf{Y} \mid \mathbf{U}, \Box A \longrightarrow \mathbf{V} \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k & \mathbf{T}\ \text{Rule} \\ \hline
\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X}, \Box A \longrightarrow \mathbf{Y} \mid \Box A \longrightarrow \mid \mathbf{U} \longrightarrow \mathbf{V} \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k & \text{Modal Splitting} \\ \hline
\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X}, \Box A \longrightarrow \mathbf{Y} \mid \mathbf{X}, \Box A \longrightarrow \mathbf{Y} \mid \mathbf{U} \longrightarrow \mathbf{V} \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k & \text{Weakening} \\ \hline
\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X}, \Box A \longrightarrow \mathbf{Y} \mid \mathbf{U} \longrightarrow \mathbf{V} \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k & \text{Repetition}
\end{array}$$

Showing that the π rules are derived ones is simpler, and does not make use of the Modal Splitting rule. Here is the verification for the first one.

$$\begin{array}{cl}
\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X}, \Diamond A \longrightarrow \mathbf{Y} \mid A \longrightarrow \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k & \\ \hline
\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X}, \Diamond A \longrightarrow \mathbf{Y} \mid \Diamond A \longrightarrow \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k & \mathbf{K4}\ \text{Rule} \\ \hline
\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X}, \Diamond A \longrightarrow \mathbf{Y} \mid \mathbf{X}, \Diamond A \longrightarrow \mathbf{Y} \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k & \text{Weakening} \\ \hline
\mathcal{C}_1 \mid \cdots \mid \mathcal{C}_n \mid \mathbf{X}, \Diamond A \longrightarrow \mathbf{Y} \mid \mathcal{D}_1 \mid \cdots \mid \mathcal{D}_k & \text{Repetition}
\end{array}$$

Finally, Figure 10 displays a hypersequent proof of $P \supset \Box\Diamond P$, in which use is made of some of the derived rules. Note that it begins with a hypersequent axiom, since $P \longrightarrow \Box\Diamond P, P$ is a sequent axiom.

$$\begin{array}{cl}
P \longrightarrow \Box\Diamond P, P \mid \longrightarrow \Diamond P & \\ \hline
P \longrightarrow \Box\Diamond P \mid \longrightarrow \Diamond P & \text{Derived } \nu \text{ Rule} \\ \hline
P \longrightarrow \Box\Diamond P & \text{Derived } \pi \text{ Rule} \\ \hline
\longrightarrow P \supset \Box\Diamond P & \text{Implication Rule}
\end{array}$$

Figure 10. A Hypersequent Proof

8.3 Soundness and Completeness

Equations (8) in Section 7.1 define a translation from sequents into formulas. Following [3] this is extended to hypersequents as follows.

$$[\mathcal{C}_1 \mid \mathcal{C}_2 \mid \cdots \mid \mathcal{C}_n]^f = \Box\mathcal{C}_1^f \vee \Box\mathcal{C}_2^f \vee \ldots \vee \Box\mathcal{C}_n^f \tag{10}$$

Suppose we could show all hypersequents that are provable in the **S5** system just presented have translates that are valid in **S5**. If the formula X were provable, there would be a hypersequent proof ending in the single-component hypersequent $\longrightarrow X$, and its hypersequent translate is $\Box(\top \supset X)$, which would be valid. But if $\Box(\top \supset X)$ is **S5** valid, so is X. Consequently, showing soundness for the **S5** hypersequent calculus comes down to showing all provable hypersequents have **S5** valid translates.

It is easy to see that the translation of each hypersequent axiom is **S5** valid. So if we can show that each of the hypersequent rules preserves validity of translation, we have soundness. This is straightforward for most of the hypersequent calculus rules. The only one needing serious work is the modal splitting rule. Let us say that component $\mathcal{C}$ is $\Box\mathbf{X}, \Diamond\mathbf{Y}, \mathbf{Z} \longrightarrow \Box\mathbf{U}, \Diamond\mathbf{V}, \mathbf{W}$, and that $\Box\mathbf{X}, \Diamond\mathbf{Y} \longrightarrow \Box\mathbf{U}, \Diamond\mathbf{V}$ and $\mathbf{Z} \longrightarrow \mathbf{W}$ is a modal splitting of it. (I write $\Box\mathbf{X}$ to denote the list of formulas in $\mathbf{X}$ with $\Box$ prefixed to each, and similarly for $\Diamond\mathbf{Y}$.) It is enough to show the **S5** validity of the following.

$$\Box[\Box\mathbf{X}, \Diamond\mathbf{Y}, \mathbf{Z} \longrightarrow \Box\mathbf{U}, \Diamond\mathbf{V}, \mathbf{W}]^f \supset \{\Box[\Box\mathbf{X}, \Diamond\mathbf{Y} \longrightarrow \Box\mathbf{U}, \Diamond\mathbf{V}]^f \vee \Box[\mathbf{Z} \longrightarrow \mathbf{W}]^f\}$$

Writing this out in full, we get the following formula.

$$\begin{aligned}\Box[(\bigwedge_i \Box X_i \wedge \bigwedge_i \Diamond Y_i \wedge \bigwedge_i Z_i) \supset (\bigvee_i \Box U_i \vee \bigvee_i \Diamond V_i \vee \bigvee_i W_i)] \supset \\ \{\Box[(\bigwedge_i \Box X_i \wedge \bigwedge_i \Diamond Y_i) \supset (\bigvee_i \Box U_i \vee \bigvee_i \Diamond V_i)] \vee \Box[\bigwedge_i Z_i \supset \bigvee_i W_i]\}\end{aligned}$$

Despite the intimidating appearance of this, it has a simple proof using the Simple Tableau System for **S5** given in Section 6.3. I'll leave this to the reader.

A direct proof of completeness can be found in [3], but for us it is easier to show that Simple **S5** Tableau System proofs can be translated into hypersequent proofs, and then rely on tableau completeness. In (9) a mapping from finite sets of signed formulas to sequents was given. Now that is used to define a mapping from finite sets of *prefixed* signed formulas to hypersequents as follows. For a finite set S of prefixed signed formulas (using positive integers as prefixes):

$$S^s = \mathcal{C}_1 \mid \mathcal{C}_2 \mid \cdots \mid \mathcal{C}_n \text{ where } \mathcal{C}_k = \{\mathcal{X} \mid k\,\mathcal{X} \in S\}^s \tag{11}$$

Now a version of Lemma 8 can be proved for the current set-up.

LEMMA 10. *Let S be a finite set of prefixed signed formulas. If there is a closed tableau for S using the Simple* **S5** *Tableau rules of Table 16 then the hypersequent S^s is provable using the rules of this section.*

Proof. As with Lemma 8 the proof is by an induction on the depth of S, where the depth is the smallest number d such that there is a closed Simple **S5** Tableau for S with d tableau rule applications. The argument for the classical connectives is just the hypersequent analog of what we did before. I'll consider one of the modal cases, and leave the others to you.

Suppose S closes with depth d, the result is known for sets that close with depth less than d, the prefixed formula $n\,T\,\Box A \in S$, and there is a d-step closed tableau for S in which the first rule application is to this prefixed formula, adding the formula $k\,T\,A$ to the tableau branch, where k already occurs (so it must be in S). Of course there is a closed tableau for $S \cup \{k\,T\,A\}$ with fewer than d rule applications. From the induction

hypothesis, the hypersequent image of this set must have a hypersequent proof. By definition,

$$[S \cup \{k\,T\,A\}]^s = \cdots \mid \mathbf{X}, \Box A \longrightarrow \mathbf{Y} \mid \cdots \mid \mathbf{U}, A \longrightarrow \mathbf{V} \mid \cdots$$

where $\mathbf{X}, \Box A$ and $\mathbf{Y}$ are the lists of members of $S \cup \{k\,T\,A\}$ with a prefix of n and signs of T and F respectively, and $\mathbf{U}, A$ and $\mathbf{V}$ are the members with a prefix of k and signs of T and F respectively. But then, using one of the Derived ν Rules, the hypersequent

$$\cdots \mid \mathbf{X}, \Box A \longrightarrow \mathbf{Y} \mid \cdots \mid \mathbf{U} \longrightarrow \mathbf{V} \mid \cdots$$

will also be provable, and this is S^s. ❑

PROPOSITION 11. *The* **S5** *hypersequent calculus is complete.*

Proof. If X is **S5** valid, there is a closed Simple **S5** Tableau for $\{1\,F\,X\}$. By the Lemma, there must be a hypersequent proof of the hypersequent whose only component is $\longrightarrow X$, and so X has a hypersequent proof. ❑

In a clear sense, the hypersequent calculus for **S5** (with the derived rules) bears the same relationship to the Simple **S5** Tableau System that the sequent calculi of Section 7 have to destructive modal tableau systems.

9 LOGICS OF KNOWLEDGE

Up to now only mono-modal logics have been considered. Things become more interesting when several modal operators are combined in a single logic. This is the first of several sections in which we examine such multi-modal logics, and is the simplest of these sections. Most proof methods adapt to multi-modal logics with varying degrees of difficulty, but from now on I will concentrate almost exclusively on prefixed tableaus. These generally make the jump to multi-modal logics with a certain grace. One natural multi-modal logic is a logic of knowledge, first investigated in detail in [36]; also see [15] for a more recent account.

9.1 A Basic Logic of Knowledge

In a logic of knowledge there is a set, usually finite, of *knowers* $\{k_1, k_2, \ldots, k_m\}$. Informally I'll use a, b, ... to range over the set of knowers. For each knower a there is a corresponding modal operator, K_a. We read K_aX as "a knows X." Each K_a is like $\Box$ from earlier sections. It is not as common to have dual modal operators, but I will—I'll use $\overline{K}_a$ for the operator dual to K_a, that is, $\neg K_a \neg$. A dual knowledge operator is like $\Diamond$ from earlier sections. Informally, $\overline{K}_aX$ can be read, "X is compatible with what a knows."

Properties of actual knowledge are difficult to specify. We might know X and also $X \supset Y$, but not know Y simply because we never thought about it. What standard logics of knowledge capture is not actual knowledge, but potential knowledge—what one is entitled to know. The switch to potential knowledge means we drop all considerations of complexity—we potentially could know a tautology with 10^{100} symbols, for instance. But the switch to an idealized point of view does simplify the theory. It is rather easy

to see that, under such an assumption, a knowledge modality should be a normal modal operator. But, what else should be required?

Since one cannot know something that is false, we would want $K_aX \supset X$, the T axiom. On the other hand belief (also idealized) is somewhat similar to knowledge, and if it is belief we are trying to examine, we would not want this axiom. Often a certain degree of introspection is assumed—if one knows something, that fact is also known—that is $K_aX \supset K_aK_aX$. Thus we might want knowledge operators to obey the $S4$ conditions. Further, negative introspection is also often assumed—if one does not know something, it is known that it is not known—that is $\neg K_aX \supset K_a\neg K_aX$. All these together make a knowledge operator obey the $S5$ conditions. But keep in mind that lesser, or different, assumptions may be appropriate in particular cases.

The semantics for a logic of knowledge is simple—a frame is a structure $\langle \mathcal{G}, \mathcal{R}_{k_1}, \mathcal{R}_{k_2}, \ldots, \mathcal{R}_{k_m} \rangle$ where $\mathcal{G}$ is the usual set of possible worlds, and we have an accessibility relation for each knower. A model is based on such a frame in the standard way, by specifying which propositional letters are true at which worlds. If $\mathcal{M}$ is such a model, truth at possible worlds is defined in the usual way, but with the modal condition: $\mathcal{M}, w \Vdash K_aX$ if and only if $\mathcal{M}, w' \Vdash X$ for each $w' \in \mathcal{G}$ with $w\mathcal{R}_aw'$. Often each $\mathcal{R}_a$ is taken to be an equivalence relation, corresponding to $S5$, but other assumptions can be made, and they do not need to be the same for each knower.

An axiomatic treatment of a logic of knowledge is quite straightforward. One simply assumes the appropriate modal axioms (and necessitation rule) for each K_a. I will skip further discussion. Various other proof systems can be adapted to logics of knowledge, but prefixed tableau systems are most 'practical,' and quite natural. They are all that will be covered here. We now take a prefix to be a sequence $1.n_1a_1.n_2a_2.\ldots$ where, except for the first item, 1, each term consists of a positive integer, n, and a knower, a. The idea is, $\sigma.na$ is intended to designate a world that is accessible from the world σ designates, via the accessibility relation for knower a. The propositional connective tableau rules are as they were in the mono-modal case. Before stating the new modal rules, the earlier nu/pi notation must be modified, since it does not keep track of which knowledge operator we are dealing with. This is done in Table 17.

ν^a	ν_0^a	π^a	π_0^a
$T\,K_aX$	$T\,X$	$T\,\overline{K}_aX$	$T\,X$
$F\,\overline{K}_aX$	$F\,X$	$F\,K_aX$	$F\,X$

Table 17. Multi-Modal Nu and Pi Formulas

In Table 13, tableau rules for mono-modal **K** were given. In the multi-modal setting these rules are replaced by those in Table 18. Likewise the mono-modal tableau rules of Table 14 must be replaced with those in Table 19.

$$\frac{\sigma\,\nu^a}{\sigma.na\,\nu_0^a} \qquad \frac{\sigma\,\pi^a}{\sigma.na\,\pi_0^a}$$

for $\sigma.na$ used   for $\sigma.na$ new

Table 18. Multi-Modal Tableau Rules for **K**

$$T^a \quad \frac{\sigma\,\nu^a}{\sigma\,\nu_0^a} \qquad\qquad 4r^a \quad \frac{\sigma.na\,\nu^a}{\sigma\,\nu^a}$$

$$4^a \quad \frac{\sigma\,\nu^a}{\sigma.na\,\nu^a} \qquad\qquad D^a \quad \frac{\sigma\,T\,K_aX}{\sigma\,T\,\overline{K}_aX} \quad \frac{\sigma\,F\,\overline{K}_aX}{\sigma\,F\,K_aX}$$

$$B^a \quad \frac{\sigma.na\,\nu^a}{\sigma\,\nu_0^a}$$

For prefixes σ and $\sigma.na$ already occurring on the tableau branch:

Table 19. Multi-Modal Tableau Rules

To illustrate how this works, Figure 11 displays a derivation using these rules. In it we have a logic of knowledge with two knowers, a and b. It is assumed that the T and $4r$ rules apply to K_b while only the K rules apply to K_a. The figure presents a derivation of $K_a\neg K_b(X \supset Y)$ from $K_a\neg K_b\neg K_b\neg Y$ and K_aX. In the tableau, 4 is from 3 by π^a; 5 is from 4 by negation; 6 is from 5 by T; 7 is from 2 and 8 is from 1 by ν^a; 9 is from 8 by negation; 10 is from 9 by π^b; 11 is from 10 by negation; 12 is from 11 by $4r^b$; 13 is from 12 by T; 14 is from 13 by negation; and 15 and 16 are from 6 by β.

I'll omit proofs of soundness and completeness. They are quite straightforward extensions of the mono-modal proofs given earlier. This is so for both the tableau and the axiomatic treatments.

Finally, as formulated above, knowers were independent beings. One might want to consider relationships between them such as, for instance, that b knows everything that a knows, or that if a knows something, b knows that a knows it. The most straightforward way of handling such things is to formulate them as axiom schemes. For instance, the first condition gives us the scheme $K_aX \supset K_bX$ and the second gives us $K_aX \supset K_bK_aX$. These can be added to an axiomatic formulation of a basic logic of knowledge, or can be taken as global assumptions in a tableau treatment. In many cases such assumptions can be reformulated as tableau rules too. For instance, the two conditions just mentioned correspond to the following two tableau rules.

$$\frac{\sigma\,\nu^a}{\sigma.nb\,\nu_0^a} \qquad\qquad \frac{\sigma\,\nu^a}{\sigma.nb\,\nu^a}$$

for $\sigma.nb$ used, corresponding to $K_aX \supset K_bX$ (left rule); for $\sigma.nb$ used, corresponding to $K_aX \supset K_bK_aX$ (right rule).

Conditions relating knowers are infinitely varied. I will not attempt to provide a general theory for them, but leave it to you to deal with them on a case-by-case basis.

9.2 Common Knowledge

A formula X is common knowledge among a group of knowers if X is true, everybody knows X, everybody knows that everybody knows X, and so on. If our knowers are,

$$
\begin{array}{ll}
1 & T\,K_a\neg K_b\neg K_b\neg Y \quad 1.\\
1 & T\,K_a X \quad 2.\\
1 & F\,K_a\neg K_b(X\supset Y) \quad 3.\\
1.1a & F\,\neg K_b(X\supset Y) \quad 4.\\
1.1a & T\,K_b(X\supset Y) \quad 5.\\
1.1a & T\,X\supset Y \quad 6.\\
1.1a & T\,X \quad 7.\\
1.1a & T\,\neg K_b\neg K_b\neg Y \quad 8.\\
1.1a & F\,K_b\neg K_b\neg Y \quad 9.\\
1.1a.1b & F\,\neg K_b\neg Y \quad 10.\\
1.1a.1b & T\,K_b\neg Y \quad 11.\\
1.1a & T\,K_b\neg Y \quad 12.\\
1.1a & T\,\neg Y \quad 13.\\
1.1a & F\,Y \quad 14.
\end{array}
$$

$$
\begin{array}{ccc}
1.1a\,F\,X \quad 15. & \qquad\qquad & 1.1a\,T\,Y \quad 16.
\end{array}
$$

Figure 11. Logic of Knowledge Tableau Example

say, $k_1, \ldots, k_n$, let us abbreviate $K_{k_1}\varphi \wedge \ldots \wedge K_{k_n}\varphi$ by $E\varphi$, and read it "everybody knows φ." Then common knowledge of X is, informally, the infinite conjunction $X \wedge EX \wedge EEX \wedge EEEX \wedge \ldots$, which we can represent as $\mathcal{C}X$. Of course this is not a real formula, though it can serve for motivation. Semantically, we want $\mathcal{C}X$ to be true at a state if X is true at every *reachable* state, where a state is reachable if there is some path to it along which each state is accessible from the previous one using any one of the accessibility relations $\mathcal{R}_{k_i}$. This seems simple enough, but capturing common knowledge axiomatically involves a fixpoint axiom and a rule of inference that is not obvious, [15]. Common knowledge cannot be captured at all by a conventional cut-free tableau system, [1], and it seems unlikely that it can be captured by a prefixed tableau system either. However, common knowledge applications often fall into two categories: how is common knowledge obtained, and how is common knowledge used given that it has been obtained. Many problems and puzzles involve only the latter, and this is relatively simple. In terms of the illegal equivalence $\mathcal{C}X \equiv X \wedge EX \wedge EEX \wedge EEEX \wedge \ldots$, obtaining common knowledge is the right-left implication, which has an infinitary antecedent, but using common knowledge involves the left-right implication, and this can be replaced with the legal, though infinite, list $\mathcal{C}X \supset X$, $\mathcal{C}X \supset EX$, $\mathcal{C}X \supset EEX$ .... In terms of tableaus using, though not obtaining, common knowledge is captured by rules that can deal with occurrences of $\mathcal{C}$ in negative positions in formulas being proved (which become positive positions when the formula is signed with F to begin a tableau proof). The rules are in Table 20. They are sound, but not complete, though they are complete with respect to a generalized version of common knowledge. I do not go into this here—see [2] for an investigation of a related system.

To show how these rules work, I'll make use of the familiar muddy children puzzle. There are, say, three children sitting in a circle, call them a, b, and c. Each can see the

$$\frac{\sigma\, T\, \mathcal{C}X}{\sigma.na\, T\, \mathcal{C}X} \qquad \frac{\sigma\, T\, \mathcal{C}X}{\sigma\, T\, X}$$

for $\sigma.na$ used

Table 20. Common Knowledge Tableau Rules

foreheads of the others, but not their own. Their legal guardian and designated puzzle person puts a spot of mud on each of their foreheads. At this point each child knows that the other two have muddy foreheads, but does not know the status of their own. Then the puzzle person announces, "at least one of you has a muddy forehead." Note that they already knew this—the effect of the announcement is to make the statement common knowledge. Then they are asked if they know whether or not their forehead is muddy. They do not, and say so. They are asked again, with the same result. They are asked a third time, and all know their forehead is muddy. The problem is to account for this.

To formalize this, let A have the intended meaning, the forehead of a is muddy, and similarly for B and C. At the start, each knows the set-up, so it is common knowledge that each knows the status of the others foreheads. That is, we have the following.

$$\mathcal{C}(K_aB \vee K_a\neg B) \tag{12}$$
$$\mathcal{C}(K_aC \vee K_a\neg C) \tag{13}$$
$$\mathcal{C}(K_bA \vee K_b\neg A) \tag{14}$$
$$\mathcal{C}(K_bC \vee K_b\neg C) \tag{15}$$
$$\mathcal{C}(K_cA \vee K_c\neg A) \tag{16}$$
$$\mathcal{C}(K_cB \vee K_c\neg B) \tag{17}$$

Let S be the formula asserting that someone knows the status of their forehead: $K_aA \vee K_a\neg A \vee K_bB \vee K_b\neg B \vee K_cC \vee K_c\neg C$. Also let $P \not\supset Q$ abbreviate $\neg(P \supset Q)$. Now, at the start it is announced that someone has a muddy forehead, $A \vee B \vee C$. Everybody hears the announcement and sees that everyone heard, and so common knowledge is obtained: $\mathcal{C}(A \vee B \vee C)$. Then it is asked if anyone knows the status of their forehead, and they do not, so $\mathcal{C}(A \vee B \vee C) \not\supset S$. Since everyone hears the answers, this is common knowledge for the next round, $\mathcal{C}(\mathcal{C}(A \vee B \vee C) \not\supset S)$. But still nobody knows, and so $\mathcal{C}(\mathcal{C}(A \vee B \vee C) \not\supset S) \not\supset S$, and this is common knowledge for the next round, $\mathcal{C}(\mathcal{C}(\mathcal{C}(A \vee B \vee C) \not\supset S) \not\supset S)$. This time everybody knows their forehead is, in fact muddy. This is expressed by the following.

$$\mathcal{C}(\mathcal{C}(\mathcal{C}(A \vee B \vee C) \not\supset S) \not\supset S) \supset (K_aA \wedge K_bB \wedge K_cC) \tag{18}$$

Formally, (18) is a consequence of (12) – (17) as local premises, assuming no more than T knowledge for each knower. A tableau proof can be found in Figure 12 of $\mathcal{C}(\mathcal{C}(\mathcal{C}(A \vee B \vee C) \not\supset S) \not\supset S) \supset K_aA$. In it some simple derived rules have been used. One is: conclude $\sigma\, T\, X$ and $\sigma\, F\, Y$ from $\sigma\, T\, X \not\supset Y$, which simply omits the step involving the definition of $\not\supset$. The other is: conclude $\sigma\sigma'\, T\, X$ from $\sigma\, T\, \mathcal{C}X$, where $\sigma\sigma'$ is any prefix that extends σ and occurs on the branch. This is a simple combination of the two rules given earlier for $\mathcal{C}$. In the tableau: 2 and 3 are from 1 by α; 4 is from 3 by π^a; 5 is from

2 by the derived $\mathcal{C}$ rule; 6 and 7 are from 5 by the derived $\not\supset$ rule; 8 is from 7 by α and the definition of S; 9 is a premise; 10 is from 9 by the derived $\mathcal{C}$ rule; 11 and 12 are from 10 by β; 13 is from 11 by T^c; 14 is from 8 by π^c; 15 is from 12 by ν^c; 16 is from 15 by negation; 17 is from 6 by the derived $\mathcal{C}$ rule; 18 and 19 are from 17 by the derived $\not\supset$ rule; 20 is from 19 by α and the definition of S; 21 is a premise; 22 is from 21 by the derived $\mathcal{C}$ rule; 23 and 24 are from 22 by β; 25 is from 23 by T^b; 26 is a premise; 27 is from 26 by the derived $\mathcal{C}$ rule; 28 and 29 are from 27 by β; 30 is from 28 by T^b; 31 is from 20 by π^b; 32 is from 24 by ν^b; 33 is from 32 by negation; 34 is from 29 by ν^b; 35 is from 34 by negation; and 32 is from 18 by the derived $\mathcal{C}$ rule. Closure is by 4 and 13; 16 and 25; 14 and 30; and 31, 32, 33, 35 and applications of the β rule.

In the version of the puzzle embodied in (18) the assumption was that everybody had a muddy forehead. If only two do, say a and b, those who have muddy foreheads know this on the second round. Here is an expression of this.

$$\neg C \supset [\mathcal{C}(\mathcal{C}(A \vee B \vee C) \not\supset S) \supset (K_a A \wedge K_b B)] \tag{19}$$

This, too is provable using the tableau rules given above, but now we must assume the **S5** rules for knowers a and b. On the other hand, the two children a and b can see the forehead of c, know it is not muddy, and can see each other and so know that they both know it, and so on. That is, $\neg C$ is not just true, it is common knowledge between a and b. If the antecedent $\neg C$ in (19) is replaced by this relativized common knowledge assumption, whose proper formulation I'll leave to you, **S5** rules are not needed.

The methods extend to n children with k muddy, but you doubtless get the general idea.

10 CONVERSE

Suppose we have a model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{V} \rangle$ with a single accessibility relation, but two modal operators, say $\Box$ and $\Box^{-1}$. Let us say $\Box$ has its usual interpretation, $\mathcal{M}, w \Vdash \Box X$ if $\mathcal{M}, w' \Vdash X$ for all $w' \in \mathcal{G}$ with $w\mathcal{R}w'$, but suppose $\Box^{-1}$ is understood via $\mathcal{M}, w \Vdash \Box^{-1} X$ if $\mathcal{M}, w' \Vdash X$ for all $w' \in \mathcal{G}$ with $w'\mathcal{R}w$. The difference is easy to miss: $\Box$ uses $\mathcal{R}$, but $\Box^{-1}$ uses the relation converse to $\mathcal{R}$, often written $\mathcal{R}^{-1}$. This gives us a natural multi-modal logic, but with the modalities intimately connected. One place where such things come up is *temporal logic*—future and past are converse in this sense. Another place is *CPDL*, propositional dynamic logic with converse. I don't want to get into these topics here (see [32, 35]) but if we assume a simple underlying modality, **K**, **T**, or something of this sort, a tableau system capable of dealing with converse is not complicated. I will discuss it briefly in this section—it is based on the treatment in [30]. See [6] for an axiomatic version.

The simplest way to approach the subject is to build on earlier work. Suppose we use a logic of knowledge with two knowers, a and b, and from now on identify K_a with $\Box$ and K_b with $\Box^{-1}$. Then, for starters, we want the tableau rules from Table 18, or perhaps from Table 19 too if more complex assumptions about $\Box$ are being made. Converseness is a relationship between our two knowers, of the sort discussed at the end of Section 9. It can be captured by the following tableau rules.

$$\frac{\sigma.na\ \nu^b}{\sigma\ \nu_0^b} \qquad \frac{\sigma.nb\ \nu^a}{\sigma\ \nu_0^a}$$

$1 \quad F\,\mathcal{C}(\mathcal{C}(\mathcal{C}(A \vee B \vee C) \not\supset S) \not\supset S) \supset K_a A \quad 1.$
$1 \quad T\,\mathcal{C}(\mathcal{C}(\mathcal{C}(A \vee B \vee C) \not\supset S) \not\supset S) \quad 2.$
$1 \quad F\,K_a A \quad 3.$
$1.1a\,F\,A \quad 4.$
$1.1a\,T\,\mathcal{C}(\mathcal{C}(A \vee B \vee C) \not\supset S) \not\supset S \quad 5.$
$1.1a\,T\,\mathcal{C}(\mathcal{C}(A \vee B \vee C) \not\supset S) \quad 6.$
$1.1a\,F\,S \quad 7.$
$1.1a\,F\,K_c C \quad 8.$
$1 \quad T\,\mathcal{C}(K_c A \vee K_c \neg A) \quad 9.$
$1.1a\,T\,K_c A \vee K_c \neg A \quad 10.$

$1.1a\,T\,K_c A \quad 11.$
$1.1a\,T\,A \quad 13.$

$1.1a \quad T\,K_c \neg A \quad 12.$
$1.1a.1c\,F\,C \quad 14.$
$1.1a.1c\,T\,\neg A \quad 15.$
$1.1a.1c\,F\,A \quad 16.$
$1.1a.1c\,T\,\mathcal{C}(A \vee B \vee C) \not\supset S \quad 17.$
$1.1a.1c\,T\,\mathcal{C}(A \vee B \vee C) \quad 18.$
$1.1a.1c\,F\,S \quad 19.$
$1.1a.1c\,F\,K_b B \quad 20.$
$1 \quad T\,\mathcal{C}(K_b A \vee K_b \neg A) \quad 21.$
$1.1a.1c\,T\,K_b A \vee K_b \neg A \quad 22.$

$1.1a.1c\,T\,K_b A \quad 23.$
$1.1a.1c\,T\,A \quad 25.$

$1.1a.1c\,T\,K_b \neg A \quad 24.$
$1 \quad T\,\mathcal{C}(K_b C \vee K_b \neg C) \quad 26.$
$1.1a.1c\,T\,K_b C \vee K_b \neg C \quad 27.$

$1.1a.1c\,T\,K_b C \quad 28.$
$1.1a.1c\,T\,C \quad 30.$

$1.1a.1c \quad T\,K_b \neg C \quad 29.$
$1.1a.1c.1b\,F\,B \quad 31.$
$1.1a.1c.1b\,T\,\neg A \quad 32.$
$1.1a.1c.1b\,F\,A \quad 33.$
$1.1a.1c.1b\,T\,\neg C \quad 34.$
$1.1a.1c.1b\,F\,C \quad 35.$
$1.1a.1c.1b\,T\,A \vee B \vee C \quad 32.$

Figure 12. Muddy Children Puzzle

I think we're at a point where I can omit soundness and completeness arguments. Table 21 shows a simple proof in this system, of $P \supset K_b\overline{K}_aP$. In it, 2 and 3 are from 1 by α, 4 is from 3 by π in Table 18; 5 is from 4 by a ν rule above.

$$\begin{array}{ll}
1 & F\,P \supset K_b\overline{K}_aP \quad 1. \\
1 & T\,P \quad 2. \\
1 & F\,K_b\overline{K}_aP \quad 3. \\
1.1b & F\,\overline{K}_aP \quad 4. \\
1 & F\,P \quad 5.
\end{array}$$

Table 21. Converse Modality Tableau Example

11 THE UNIVERSAL MODALITY AND THE DIFFERENCE MODALITY

Logics of knowledge combine many 'ordinary' modal operators, but there has been considerable investigation of the effects of adding special, expressive, modal operators to a standard modal logic. Two of these that are especially powerful and interesting are the *universal* modality, also called the *global* modality in [6], and the *difference* modality. Discussion of, and axiomatization for both can be found in [6].

Suppose we have a multi-modal logic **L** of the kind considered in Section 9, in which the individual modal operators are among those of Table 15, or **K** of course. That is, we have a prefixed tableau system for **L**. The *universal* modal operator, written E, is a kind of possibility operator that can be read "somewhere." That is, EX informally asserts that X holds at some possible world. Note that the accessibility relation plays no role in this. The dual necessity-like operator is written A. In a model $\mathcal{M}$ for **L** with these operators added, we want the following conditions.

$$\mathcal{M}, w \Vdash EX \iff \mathcal{M}, w' \Vdash X \text{ for some } w' \in \mathcal{M}$$
$$\mathcal{M}, w \Vdash AX \iff \mathcal{M}, w' \Vdash X \text{ for every } w' \in \mathcal{M}$$

To provide prefixed tableau rules for these operators, we need a small extension of the system in Section 9. There a prefix was a sequence of the form $1.n_1a_1.n_2a_2.\ldots$, beginning with 1. From now on prefixes can begin with any positive integer. The nu/pi definition is extended in the obvious way, in Table 22.

ν	ν_0
$T\,AX$	$T\,X$
$F\,EX$	$F\,X$

π	π_0
$T\,EX$	$T\,X$
$F\,AX$	$F\,X$

Table 22. Universal Modality Nu and Pi Formulas

The tableau rules for E and A are given in Table 23. These are to be added to the rules for the other modal operators of **L**.

Soundness and completeness proofs are not difficult, and I omit them. I make one observation, however. It is easy to see that models produced by earlier completeness

$$\frac{\sigma\,\nu}{\sigma'\,\nu_0} \qquad\qquad \frac{\sigma\,\pi}{n\,\pi_0}$$

for σ' used for n new

Table 23. Universal Modality Tableau Rules

proofs for prefixed tableau systems all have a tree structure, since that is the syntactical structure of prefixes. This is no longer the case when the universal modality is present. Instead models are more like forests—sets of trees.

The *difference* modality, also read as *elsewhere*, is a possibility-like operator, too. The formula DX informally can be read as asserting that X holds somewhere else. There is, of course, a dual operator, but since the tableau rules take a somewhat more complex form, I won't consider it. Once again, a discussion and axiomatization for this operator can be found in [6]. As to prefixed tableau rules, the addition to **L** given in Table 24 will do.

$$\frac{T\,\sigma\,DX}{T\,n\,X \;\big|\; T\,\sigma_1\,X \;\big|\; T\,\sigma_2\,X \;\big|\; \cdots \;\big|\; T\,\sigma_k\,X} \qquad\qquad \frac{F\,\sigma\,DX}{F\,\sigma'\,X}$$

In the T rule: n is a new integer, and $\sigma_1, \ldots, \sigma_k$ are all prefixes used on the branch other than σ. In the F rule: σ' is any prefix used on the branch other than σ.

Table 24. Difference Modality Tableau Rules

Figure 13 displays a sample proof using these rules, of $DDP \supset (P \vee DP)$, which is one of the axioms for D in [6]. In it, 2 and 3 are from 1 by α; 4 and 5 are from 3 by α; 6 is from 2 by the TD rule (note that at this point there are no prefixes on the branch other than 1, and 2 is new); 7 and 8 are from 6 by the TD rule (3 is new, and 1 is the only prefix on the branch other than 2); 9 is from 5 by the FD rule.

$$\begin{array}{l}
1\,F\,DDP \supset (P \vee DP) \quad 1.\\
1\,T\,DDP \quad 2.\\
1\,F\,P \vee DP \quad 3.\\
1\,F\,P \quad 4.\\
1\,F\,DP \quad 5.\\
2\,T\,DP \quad 6.
\end{array}$$

$$\begin{array}{ll}
3\,T\,P \quad 7. & \qquad 1\,T\,P \quad 8.\\
3\,F\,P \quad 9. &
\end{array}$$

Figure 13. Prefixed Difference Modality Tableau

12 WHAT ARE THE LIMITATIONS

Going beyond the logics we have looked at so far are those using a *fixpoint* construction. That is, in the semantics some modal operator is modeled by a device that involves minimizing (or maximizing) some monotone operator on the set of possible worlds. Such logics are very powerful, and very resistant to tableau methods. It is an area needing further research. Here is a brief rundown of the most common fixpoint logics.

Propositional dynamic logic is, perhaps, the original example, see [35] for a thorough treatment. In this multi-modal logic, modal operators correspond to computer programs. The semantical treatment of the *while* operator requires a fixpoint construction. From early on there was a tableau system for the logic, [51]. It is, however, of a specialized nature that so far has not lent itself well to treatment by the general methodologies of this chapter.

Propositional dynamic logic was extended to the *propositional μ-calculus* in [41], with modal operators in the language corresponding directly to fixpoint constructions. I do not know how to bring tableau methods to bear.

Finally, common knowledge was discussed above, in Section 9.2. It was made clear that the tableau rules given were only for *using* common knowledge, and could not handle its acquisition. Common knowledge is yet another example of a fixpoint modality, and another example of one for which satisfactory tableau rules do not exist.

13 QUANTIFIED MODAL LOGIC

As if propositional modal logic wasn't complicated enough, each one can be extended to a first-order version in a multiplicity of ways [28, 37, 26]. For starters, domains of quantification can be different at different worlds, or the same at all worlds, or related in ways that depend on relative accessibility. When each possible world has its own domain of quantification, one can think of these domains as representing what exists at each world. In this case quantifiers are *actualist*—they quantify over what actually exists. When the domain of quantification is the same for all worlds, one can think of the common domain as the realm of possible existents. Then quantifiers are *possibilist*. These two versions correspond to well-established philosophical positions, but this is not the place to discuss them. As it happens, constant domain, possibilist quantification is the easiest to capture using prefixed tableaus. And, as it happens, varying domain semantics can be embedded into a constant domain version easily and naturally. So, in this section I'll briefly sketch prefixed tableau systems for constant domain, possibilist versions of the propositional modal logics that were treated earlier, and then say a little about other versions of quantified modal logic.

13.1 Syntax and Semantics

We need a first-order language. I'll assume we have relation symbols of arity 1, 2, ... In the interests of simplicity, there will be no constant or function symbols. I'll also assume there is an infinite list of variables. Atomic formulas are expressions of the form $P(v_1, \ldots, v_n)$ where P is a relation symbol of arity n, and $v_1, \ldots, v_n$ are variables. Formulas are built up from atomic formulas in the usual way, using modal operators ($\Box$ and $\Diamond$ only, for the time being), propositional connectives, and quantifiers. I'll take both

$\forall$ and $\exists$ as primitive in this section. I'll assume the notion of free and bound variable is understood, and so the notion of closed formula, or sentence.

A *constant domain model* is a structure $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$ where: $\langle \mathcal{G}, \mathcal{R} \rangle$ is a frame as before; $\mathcal{D}$ is a non-empty set, the domain of quantification; and $\mathcal{I}$ is an interpretation, assigning to each n-ary relation symbol P and each possible world w an n-place relation $\mathcal{I}(P, w)$ on $\mathcal{D}$. We are primarily interested in the behavior of sentences, but formulas with free variables come into things, so we need one more piece of machinery. A *valuation* in a model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$ is a mapping v from the set of variables to the domain, $\mathcal{D}$, of the model.

The chief semantic notion is symbolized $\mathcal{M}, w \Vdash_v X$ and is read: formula X is true at world w of model $\mathcal{M}$ with respect to valuation v. The definition follows. In it, a valuation v' is an x-variant of a valuation v if the two valuations agree on all variables except possibly x.

Atomic $\mathcal{M}, w \Vdash_v P(x_1, \ldots, x_n)$ if and only if $\langle v(x_1), \ldots, v(x_n) \rangle \in \mathcal{I}(P, w)$

Negation $\mathcal{M}, w \Vdash_v \neg X$ if and only if not-$\mathcal{M}, w \Vdash_v X$

Propositional Connectives $\mathcal{M}, w \Vdash_v X \wedge Y$ if and only if $\mathcal{M}, w \Vdash_v X$ and $\mathcal{M}, w \Vdash_v Y$, and similarly for the other connectives.

Necessity $\mathcal{M}, w \Vdash_v \Box X$ if and only if $\mathcal{M}, w' \Vdash_v X$ for every $w' \in \mathcal{G}$ with $w \mathcal{R} w'$.

Possibility $\mathcal{M}, w \Vdash_v \Diamond X$ if and only if $\mathcal{M}, w' \Vdash_v X$ for some $w' \in \mathcal{G}$ with $w \mathcal{R} w'$.

Universal Quantifier $\mathcal{M}, w \Vdash_v (\forall x)\varphi$ if and only if $\mathcal{M}, w \Vdash_{v'} \varphi$ for every valuation v' that is an x-variant of v.

Existential Quantifier $\mathcal{M}, w \Vdash_v (\exists x)\varphi$ if and only if $\mathcal{M}, w \Vdash_{v'} \varphi$ for some valuation v' that is an x-variant of v.

As might be expected, if X has no free variables, $\mathcal{M}, w \Vdash_v X$ for some valuation v if and only if $\mathcal{M}, w \Vdash_v X$ for every valuation v. Consequently we can speak of the truth or falsity of a sentence at a world in a model without mentioning the valuation. I'll say a sentence X is **L** valid, where **L** is some normal modal logic determined by a class of frames, if X is true at every world of every model $\langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$ such that $\langle \mathcal{G}, \mathcal{R} \rangle$ is an **L** frame, for every non-empty domain $\mathcal{D}$ and every interpretation $\mathcal{I}$.

13.2 Constant Domain Tableaus

The goal is to extend the tableau systems of Section 6 to take constant domain quantification into account. As it happens, this is accomplished easily by a combination of modal rules and standard tableau quantification rules, [22, 57].

Tableau proofs will be of sentences only. As usual in treatments of first-order logic, there will be a version of existential instantiation—if $(\exists x)P(x)$ is true then $P(x)$ is true for some value of x, so we introduce a symbol intended to designate such a value. For this purpose there is a second list of variables, disjoint from the first. These new variables are called *parameters*. Formulas with parameters as free variables play a role in proofs, but they have no part in what we are trying to prove. From now on, by *sentence* I mean a closed formula none of whose variables are parameters.

Let **L** be one of the propositional modal logics for which a prefixed tableau system was given in Section 6. For a constant domain, first-order version of this we use the propositional rules for **L** from before, and add to them rules for quantifiers. To make it easier to state the new rules, two new formula classifications are introduced, following [57]. In giving them, I'll adopt the following convention. If I write $\varphi(x)$ it is intended to represent a formula in which at most x may occur free; then a subsequent occurrence of $\varphi(p)$ represents the result of substituting the parameter p for every free occurrence of x in $\varphi(x)$. Note that since parameters cannot be quantified, we needn't worry about 'accidental' capture of p in such a substitution. Now, Table 25 contains the new formula cases, and Table 26 gives the new tableau rules. The γ rule allows any parameter to be used, while the δ rule requires a parameter that is new to the branch.

γ	$\gamma(p)$
$T\,(\forall x)\varphi(x)$	$T\,\varphi(p)$
$F\,(\exists x)\varphi(x)$	$F\,\varphi(p)$

δ	$\delta(p)$
$T\,(\exists x)\varphi(x)$	$T\,\varphi(p)$
$F\,(\forall x)\varphi(x)$	$F\,\varphi(p)$

Table 25. Gamma and Delta Formulas

$$\frac{\sigma\,\gamma}{\sigma\,\gamma(p)}\ \text{for any } p \qquad\qquad \frac{\sigma\,\delta}{\sigma\,\delta(p)}\ \text{for } p \text{ new}$$

Table 26. Quantifier Rules

Figure 14 displays a proof of $(\forall x)\Box P(x) \supset \Box(\forall x)P(x)$, an instance of the *Barcan formula*, using the propositional **K** rules. In it, 2 and 3 are from 1 by α; 4 is from 3 by π; 5 is from 4 by δ (p is a new parameter); 6 is from 2 by γ; 7 is from 6 by ν.

$$\begin{array}{ll}
1 & F\,(\forall x)\Box P(x) \supset \Box(\forall x)P(x) \quad 1. \\
1 & T\,(\forall x)\Box P(x) \quad 2. \\
1 & F\,\Box(\forall x)P(x) \quad 3. \\
1.1 & F\,(\forall x)P(x) \quad 4. \\
1.1 & F\,P(p) \quad 5. \\
1 & T\,\Box P(p) \quad 6. \\
1.1 & T\,P(p) \quad 7.
\end{array}$$

Figure 14. Prefixed **K** Barcan Formula Proof

13.3 Soundness and Completeness

By now several soundness and completeness arguments for tableau systems have been given. I think we are at the point where we can be somewhat more terse. A set S of quantified formulas is *satisfiable* if there is a model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$, a valuation v, and a mapping n from the prefixes in S to possible worlds in $\mathcal{M}$, such that if $\sigma\,T\,X \in S$ then $\mathcal{M}, n(\sigma) \Vdash_v X$ and if $\sigma\,F\,X \in S$ then $\mathcal{M}, n(\sigma) \not\Vdash_v X$. A tableau branch is satisfiable if

the set of formulas on it is, and a tableau is satisfiable if some branch is. A proof that tableau rule applications preserve satisfiability can be left to you. The new thing here is the quantifier rules, and they are handled exactly as with standard classical tableau arguments. Once this is established, soundness is by the usual argument.

Completeness is a mix of modal and classical techniques. To keep the discussion uncluttered, I'll assume the underlying logic is **K**; adapting the argument to other logics that have prefixed tableau systems is straightforward.

For starters, the Lindenbaum construction of (1) and (6) needs to be 'doubly Henkinized,' to take care of both prefixes and quantifiers. I'll work with formulas that can contain free variables, but they must all be parameters. Call a set S of prefixed signed formulas $\exists$-*complete* provided, if $\sigma\,\delta \in S$ then for some parameter p, $\sigma\,\delta(p) \in S$. The notion of π-completeness was defined in Section 6.2, as was the notion of omitting infinitely many integers. S omits infinitely many parameters if the set of parameters not occurring in formulas of S is infinite. We still say a set S is **K**-consistent if no **K**-tableau for a finite part of S is closed (though now the tableau rules include those for quantifiers). Every **K**-consistent set S of prefixed formulas that omits infinitely many integers and omits infinitely many parameters can be extended to a set that is maximally **K**-consistent, π-complete, and $\exists$-complete. This can be done via the following modification of the earlier construction, (6).

Double Lindenbaum-Henkin Construction Suppose S is a **K**-consistent set of prefixed sentences. Enumerate the prefixed signed formulas of the language, $\sigma_1\,\mathcal{X}_1$, $\sigma_2\,\mathcal{X}_2$, $\ldots$, whose only free variables are parameters, and define the following sequence of sets.

$$S_1 = S$$

$$S_{n+1} = \begin{cases} S_n \cup \{\sigma_n\,\mathcal{X}_n\} & \text{if } \mathbf{K}\text{-consistent and } \mathcal{X}_n \text{ is not } \pi \text{ or } \delta \\ S_n \cup \{\sigma_n\,\pi, \sigma_n.k\,\pi_0\} & \text{if } \mathbf{K}\text{-consistent, } \mathcal{X}_n \text{ is } \pi\text{, and } \sigma_n.k \text{ is new} \\ S_n \cup \{\sigma_n\,\delta, \sigma_n\,\delta(p)\} & \text{if } \mathbf{K}\text{-consistent, } \mathcal{X}_n \text{ is } \delta, \\ & \quad \text{and } p \text{ is a new parameter} \\ S_n & \text{otherwise} \end{cases} \tag{20}$$

I'll leave it to you to check that $\cup_n S_n$ is maximally **K**-consistent, and both π-complete and $\exists$-complete. Now tableau completeness follows easily. Suppose the sentence X is not provable using the constant domain **K**-tableau rules. Then $\{1\,F\,X\}$ is **K**-consistent, omits infinitely many parameters (all of them), and omits infinitely many integers. Extend it to a maximally **K**-consistent, π-complete, $\exists$-complete set S. Define a model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I} \rangle$ as follows. $\mathcal{G}$ is the set of prefixes in S. For σ and τ in $\mathcal{G}$ set $\sigma\mathcal{R}\tau$ if $\tau = \sigma.n$ for some n. This much is exactly as in Section 6.2. Let $\mathcal{D}$ be the set of parameters. For an n-place relation symbol P, $\mathcal{I}(P, \sigma) = \{\langle p_1, \ldots, p_n \rangle \mid \sigma\,T\,P(p_1, \ldots, p_n) \in S\}$.

Now that we have a model, we need a version of the truth lemma, which for propositional prefixed tableaus took the form of (7). The version we need now is the following.

$$\begin{aligned} \sigma\,T\,Z \in S &\Longrightarrow \mathcal{M}, \sigma \Vdash_v Z \text{ for some } v \\ \sigma\,F\,Z \in S &\Longrightarrow \mathcal{M}, \sigma \nVdash_v Z \text{ for some } v \end{aligned} \tag{21}$$

This is proved by induction on Z. I leave the argument to you. But the consequence

is that the unprovable sentence X is falsified in this model, at world 1. Thus we have completeness for quantified **K**. Extending the argument to other logics is an exercise.

13.4 Variations

In the previous section I presented, in some detail, prefixed tableaus for constant domain **K**, and in outline versions for other constant domain logics. The other extreme is to have a semantics with varying domains, with no special conditions imposed on these domains. Thus, domains from different possible worlds might be disjoint, overlap, coincide, whatever. Recall, varying domain semantics is appropriate for *actualist quantification*, while constant domain semantics corresponds to *possibilist quantification*.

One way of developing prefixed tableaus for varying domain semantics is to introduce a different set of parameters for each prefix. This approach is worked out in detail in [26]. There is, however, a much simpler approach, one that builds directly on what has already been done here. Let us introduce a special one-place relation symbol, E, and read $\mathsf{E}(x)$ informally as *x actually exists*. And let us also introduce relativized quantifiers: $(\forall^{\mathsf{E}} x)\varphi$ abbreviates $(\forall x)[\mathsf{E}(x) \supset \varphi]$, and $(\exists^{\mathsf{E}} x)\varphi$ abbreviates $(\exists x)[\mathsf{E}(x) \wedge \varphi]$. If we use these relativized quantifiers, with the prefixed tableau rules given above, the result corresponds exactly to varying domain semantics. Loosely speaking, it is the interpretation of the E predicate, which can change from world to world, that gives us the effect of varying domains.

What was just said is not entirely accurate. One generally takes domains of quantification to be non-empty. The approach outlined above does not impose such a requirement, since the interpretation of E at a world might, in fact, be empty. But there is a simple way around this. Recall how local and global premises were used in propositional prefixed tableaus—see the end of Section 6.1. The same thing works even with quantifier rules added. So, if we want only non-empty domains, we just take $(\exists x)\mathsf{E}(x)$ as a global premise, which means $\sigma\, T\, (\exists x)\mathsf{E}(x)$ can be added to any tableau branch, for any prefix already present.

Other conditions are sometimes imposed on varying domains. *Monotonicity* is a common assumption: if $w\mathcal{R}w'$ then the domain associated with world w is a subset of that associated with world w'. *Anti-monotonicity* is also used occasionally, if $w\mathcal{R}w'$ then the domain associated with w' is a subset of that associated with w. Both can easily be incorporated into the present approach. For monotonicity, take $(\forall x)[\mathsf{E}(x) \supset \Box\mathsf{E}(x)]$ as a global premise. For anti-monotonicity, use $(\forall x)[\Diamond\mathsf{E}(x) \supset \mathsf{E}(x)]$ as a global premise.

As an example, in Figure 14 there is a proof of the Barcan formula, using possibilist quantification. Try and prove the corresponding actualist quantification version, $(\forall^{\mathsf{E}} x)\Box P(x) \supset \Box(\forall^{\mathsf{E}} x)P(x)$. You can't, with or without $(\exists x)\mathsf{E}(x)$ as a global premise. But it is provable if we assume anti-monotonicity, taking $(\forall x)[\Diamond\mathsf{E}(x) \supset \mathsf{E}(x)]$ as a global premise. A proof can be found in Figure 15. In this, 2 and 3 are from 1 by α; 4 is from 3 by π; $4'$ is 4 unabbreviated; 5 is from $4'$ by δ; 6 and 7 are from 5 by α; $2'$ is 2 unabbreviated; 8 is from $2'$ by γ; 9 and 10 are from 8 by β; 11 is a global premise; 12 is from 11 by γ; 13 and 14 are from 12 by β; 15 is from 13 by ν; 16 is from 10 by ν. Closure is by 6 and 15, 9 and 14, 7 and 16.

In the same way that possibilist quantifiers and quantification rules were added to mono-modal logics, they can be added to a multi-modal logic such as one of the logics of knowledge discussed in Section 9. Indeed, one could even introduce an actually exists

predicate for each knower, E_a, and give each modality its own domain of quantification at each world. But I think we have taken things far enough.

$$
\begin{array}{lll}
1 & F\,(\forall^{\mathsf{E}} x)\Box P(x) \supset \Box(\forall^{\mathsf{E}} x)P(x) & 1.\\
1 & T\,(\forall^{\mathsf{E}} x)\Box P(x) & 2.\\
1 & F\,\Box(\forall^{\mathsf{E}} x)P(x) & 3.\\
1.1 & F\,(\forall^{\mathsf{E}} x)P(x) & 4.\\
1.1 & F\,(\forall x)[\mathsf{E}(x) \supset P(x)] & 4'.\\
1.1 & F\,\mathsf{E}(p) \supset P(p) & 5.\\
1.1 & T\,\mathsf{E}(p) & 6.\\
1.1 & F\,P(p) & 7.\\
1 & T\,(\forall x)[\mathsf{E}(x) \supset \Box P(x)] & 2'.\\
1 & T\,\mathsf{E}(p) \supset \Box P(p) & 8.
\end{array}
$$

$$
\begin{array}{lll}
1 & F\,\mathsf{E}(p) & 9.\\
1 & T\,(\forall x)[\Diamond\mathsf{E}(x) \supset \mathsf{E}(x)] & 11.\\
1 & T\,\Diamond\mathsf{E}(p) \supset \mathsf{E}(p) & 12.
\end{array}
\qquad
\begin{array}{lll}
1 & T\,\Box P(p) & 10.\\
1.1 & T\,P(p) & 16.
\end{array}
$$

$$
\begin{array}{lll}
1 & F\,\Diamond\mathsf{E}(p) & 13.\\
1.1 & F\,\mathsf{E}(p) & 15.
\end{array}
\qquad
\begin{array}{lll}
1 & T\,\mathsf{E}(p) & 14.
\end{array}
$$

Figure 15. Anti-Monotonicity and Barcan Formula

14 CONCLUSION

We have reached the end of the chapter, but not the end of the subject. The literature on modal proof theory is vast, and there are many different approaches besides what was covered here. Here is a select list of pointers (for which I thank the reader for this chapter, Heinrich Wansing). There are higher-arity sequent systems, [7]; higher-level sequent systems, [13]; higher-dimensional sequent systems, [45]. There are display sequent systems, [4, 12, 53, 58, 60], which are particularly appropriate for temporal systems. See [61] for the relationship between these and hypersequents. In addition, there are relational proof systems, [49]; and multiple-sequent systems, [38, 39]. And for general treatments, see [59, 62].

Quantified classical logic generally admits constant and function symbols, and equality. Adding these to quantified modal logic requires choices. Are constant and function symbols to be rigid—having the same designation at each world—or non-rigid? If non-rigidity is the choice, a device called *predicate abstraction* can be added to help sort out ambiguities that arise. How does one deal with the contingent equality of the number 9 and the number of the planets, but not their synonymy? Such things can, in fact, be dealt with, and prefixed tableau systems exist that can help sort things out. See [26] for

a thorough discussion, and [23] for an abbreviated one. Also [25] contains a presentation of a rich first-order system along these lines.

After first-order comes second-order, and full type theory. This tends to get enormously complex. Tableau systems for a version of modal type theory can be found in [24].

To a certain extent I have followed my own interests in this chapter. I've tried to keep it in control, but I think my biases show. The reader should keep in mind that this is an enormous subject, and my tastes may not be that of others. Start here, don't finish here.

BIBLIOGRAPHY

[1] L. Alberucci and G. Jaeger. About cut elimination for logics of common knowledge. *Annals of Pure and Applied Logic*, 133:73–99, 2005.

[2] Sergei Artemov. Evidence-based common knowledge. Technical Report TR-200418, CUNY Program in Computer Science, 2004. *http://www.cs.gc.cuny.edu/ sartemov/publications/EBKrevised.pdf.*

[3] Arnon Avron. The method of hypersequents in the proof theory of propositional non-classical logics. In W. Hodges, M. Hyland, C. Steinhorn, and J. Truss, editors, *Logic: Foundations to Applications*, pages 1–32. Oxford Science Publications, 1996.

[4] N. D. Belnap. Display logic. *Journal of Philosophical Logic*, 11:375–417, 1982.

[5] E. W. Beth. *The Foundations of Mathematics.* North-Holland, Amsterdam, 1959.

[6] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic.* Cambridge University Press, Cambridge, UK, 2001.

[7] S. Blamey and L. Humberstone. A perspective on modal sequent logic. *Publications of the Research Institute for Mathematical Sciences, Kyoto University*, 27:763–782, 1991.

[8] George Boolos. *The Unprovability of Consistency.* Cambridge University Press, 1979.

[9] George Boolos. *The Logic of Provability.* Cambridge University Press, 1993. Paperback 1995.

[10] T. Borghuis. Interpreting modal natural deduction in type theory. In M. de Rijke, editor, *Diamonds and Defauots*, pages 67–102. Kluwer, Dordrecht, 1993.

[11] Marcello D'Agostino, Dov Gabbay, Reiner Hähnle, and Joachim Posegga, editors. *Handbook of Tableau Methods.* Kluwer, Dordrecht, 1998.

[12] S. Demri and R. Goré. Cut-free display calculi for nominal tense logics. In *Proc. Tableaux '99*, Lecture Notes in AI, pages 155–170, Berlin, 1999. Springer-Verlag.

[13] K. Došen. Sequent systems for modal logic. *Journal of Symbolic Logic*, 50:149–159, 1985.

[14] K. Došen and P. Schroeder-Heister, editors. *Substructural Logics.* Oxford University Press, Oxford, 1993.

[15] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning About Knowledge.* MIT Press, Cambridge, 1995.

[16] F. B. Fitch. *Symbolic Logic, An Introduction.* The Ronald Press, New York, 1952.

[17] Frederick Fitch. Natural deduction rules for obligation. *American Philosophical Quarterly*, 3:27–38, 1966.

[18] Melvin C. Fitting. Tableau methods of proof for modal logics. *Notre Dame Journal of Formal Logic*, 13:237–247, 1972.

[19] Melvin C. Fitting. *Proof Methods for Modal and Intuitionistic Logics.* D. Reidel Publishing Co., Dordrecht, 1983.

[20] Melvin C. Fitting. Destructive modal resolution. *Journal of Logic and Computation*, 1:83–97, 1990.

[21] Melvin C. Fitting. Basic modal logic. In Dov M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, number 1, pages 368–448. Oxford University Press, 1993.

[22] Melvin C. Fitting. *First-Order Logic and Automated Theorem Proving.* Springer-Verlag, 1996. First edition, 1990.

[23] Melvin C. Fitting. First order alethic modal logic. In Dale Jacquette, editor, *A Companion to Philosophical Logic*, chapter 27, pages 410–421. Blackwell, Malden, MA, 2002.

[24] Melvin C. Fitting. *Types, Tableaus, and Gödel's God.* Kluwer, 2002.

[25] Melvin C. Fitting. First-order intensional logic. *Annals of Pure and Applied Logic*, 127:171–193, 2004.

[26] Melvin C. Fitting and Richard Mendelsohn. *First-Order Modal Logic.* Kluwer, 1998. Paperback, 1999.

[27] Dov M. Gabbay. *Labelled Deductive Systems.* Number 33 in Oxford Logic Guides. Oxford University Press, 1996.

[28] James W. Garson. Quantification in modal logic. In Dov Gabbay and Franz Guenthner, editors, *Handbook of Philosophical Logic*, number 2, pages 249–307. D. Reidel, 1984.

[29] Gerhard Gentzen. Investigation into logical deduction. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland, 1969. Originally published as 'Untersuchungen über das logische Schliessen', in *Mathematische Zeitschrift 39* (1935), 176–210 and 405–431.

[30] G. De Giacomo and F. Massacci. Combining deduction and model checking into tableaux and algorithms for converse-PDL. *Information and Computation*, 162:117–137, 2000.

[31] Kurt Gödel. An interpretation of the intuitionistic propositional calculus. In Solomon Feferman, editor, *Kurt Gödel, Collected Works, Volume One*, pages 300–303. Oxford, 1986. Originally published as 'Eine Interpretation des intuitionistischen Aussagenkalküls', in *Ergebnisse eines mathematischen Kolloquiums*, vol 4, pp 39–40 (1933).

[32] R. Goldblatt. *Logics of Time and Computation.* CSLI Lecture Notes, Stanford, 1987.

[33] Rajeev Goré. Tableau methods for modal and temporal logics. 1998. In [11].

[34] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.

[35] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic.* MIT Press, Cambridge, 2000.

[36] Jaakko Hintikka. *Knowledge and Belief.* Cornell University Press, 1962.

[37] G. E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic.* Routledge, London, 1996.

[38] A. Indrzejczak. Generalized sequent calculus for propositional modal logics. *Logica Trianguli*, 1:15–31, 1997.

[39] A. Indrzejczak. Cut-free double sequent calculus for S5. *Logic Journal of the IGPL*, 6:505–516, 1998.

[40] S. Jàskowski. On the rules of suppositions in formal logic. *Studia Logica*, 1, 1934.

[41] Dexter Kozen. Results on the propositional μ-calculus. In E. Schmidt, editor, *Proceedings of the Ninth International Colloquium on Automata, Languages, and Programming*, Lecture Notes in Computer Science, 140, pages 348–359. Springer, 1982.

[42] Richard E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM J. Comput.*, 6:467–480, 1977.

[43] A. Leitsch. *The Resolution Calculus.* Springer Verlag, Berlin, 1997.

[44] Zbigniew Lis. Wynikanie semantyczne a wynikanie formalne (logical consequence, semantic and formal). *Studia Logica*, 10:39–60, 1960. Polish, with Russian and English summaries.

[45] A. Masini. 2-sequent calculus: a proof theory of modalities. *Annals of Pure and Applied Logic*, 58:229–246, 1992.

[46] Fabio Massacci. Strongly analytic tableaux for normal modal logics. In Alan Bundy, editor, *Proceedings of CADE 12*, volume 814 of *Lecture Notes in Artificial Intelligence*, pages 723–737, Berlin, 1994. Springer-Verlag.

[47] M. Ohnishi and K. Matsumoto. Gentzen method in modal calculi I. *Osaka Mathematical Journal*, 9:113–130, 1957.

[48] M. Ohnishi and K. Matsumoto. Gentzen method in modal calculi II. *Osaka Mathematical Journal*, 11:115–120, 1959.

[49] E. Orlowska. Relational proof systems for modal logics. In H. Wansing, editor, *Proof Theory of Modal Logic*, pages 55–77, Dordrecht, 1996. Kluwer Academic Publishers.

[50] F. Paoli. *Substructural Logics: A Primer.* Kluwer, Dordrecht, 2002.

[51] V. R. Pratt. A practical decision method for propositional dynamic logic. In *Proceedings of the Tenth Symposium on the Theory of Computation*, pages 326–337, 1978.

[52] Dag Prawitz. *Natural Deduction, A Proof-Theoretical Study.* Acta Universitatis Stockholmiensis, Stockholm Studies in Philosophy 3. Almqvist & Wiksell, Stockholm, 1965.

[53] G. Restall. *An Introduction to Substructural Logics.* Routledge, London, 2000.

[54] H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logic. In Steig Kanger, editor, *Proceedings of the Third Scandinavian Logic Symposium, Uppsala 1973*, pages 110–143, Amsterdam, 1975. North-Holland Publishing Company.

[55] D. Siemens Jr. Fitch-style rules for many modal logics. *Notre Dame Journal of Formal Logic*, 18:631–636, 1977.

[56] Craig Smoryński. *Self-reference and Modal Logic.* Universitext, Springer-Verlag, 1985.

[57] Raymond M. Smullyan. *First-Order Logic.* Springer-Verlag, Berlin, 1968. Revised Edition, Dover Press, New York, 1994.

[58] H. Wansing. Sequent systems for normal modal propositional logics. *Journal of Logic and Computation*, 4:125–142, 1994.

[59] H. Wansing, editor. *Proof Theory of Modal Logic.* Kluwer Academic Publishers, Dordrecht, 1996.

[60] H. Wansing. *Displaying Modal Logic.* Kluwer Academic Publishers, Dordrecht, 1998.

[61] H. Wansing. Translation of hypersequents into display sequents. *Journal of the IGPL*, 6:719–733, 1998.

[62] H. Wansing. Sequent systems for modal logics. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, number 8, pages 61–145. Kluwer Academic Publishers, Dordrecht, 2nd edition, 2002.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

3

COMPLEXITY OF MODAL LOGIC

Maarten Marx

1 Introduction . . . 139
  1.1 Examples of decision problems in modal logic . . . 140
  1.2 A simple and a hard problem . . . 141
  1.3 The model checking problem . . . 143
  1.4 The consequence problem . . . 145
  1.5 A tiling logic . . . 147
2 Decision algorithms . . . 148
  2.1 Selection of points . . . 149
  2.2 Filtration . . . 150
  2.3 Hintikka set elimination . . . 151
  2.4 Hintikka set elimination without constraints . . . 153
  2.5 Forcing exponentially deep paths . . . 153
  2.6 Tree automata . . . 155
  2.7 Pseudo-models . . . 158
3 Complexity . . . 164
  3.1 Computability . . . 164
  3.2 Computational complexity . . . 165
  3.3 The complexity of modal decision problems . . . 167
  3.4 Reductions . . . 168
  3.5 Tiling . . . 168
  3.6 Language design and complexity . . . 175
4 Historical notes . . . 176

1 INTRODUCTION

This chapter is a basic introduction to the field of computational complexity in modal logic. We are mostly concerned with the following question: given a formula A and a set of formulas C, does there exists a model in which all of C is true at every world and A is true at some world? In other words, is $C \models \neg A$ or $C \not\models \neg A$ the case? This is the complement of the (global) consequence problem: $C \models A$ (is A true in every model in which all of C is true at every world). The special case of the consequence problem in which C is the empty set is called the validity problem, and its complement is the satisfiability problem. For finite C, the local consequence problem reduces to the validity problem, because of the deduction theorem.

For many modal logics, these problems are decidable. Here we look at the *difficulty* of deciding them. This is the topic of the theory of computational complexity. As Wikipedia has it:

> Computational complexity theory is part of the theory of computation dealing with the resources required during computation to solve a given problem. The most common resources are time (how many steps it takes to solve a problem) and space (how much memory it takes). Complexity theory differs from computability theory, which deals with whether a problem can be solved at all, regardless of the resources required.

Standard references to this field are [27] and [32].

Organization. The current section introduces common decision problems in modal logic and derives three useful properties of modal logics. In Section 2 we discuss the basic methods of establishing decidability and complexity results for the satisfiability problem in modal logic. In Section 3 we review the basic notions of computational complexity theory and after that we reduce several tiling problems to modal satisfiability problems in order to obtain lower complexity bounds. These say roughly that —up to a polynomial— one cannot give a better algorithm for the problem at hand. Throughout the text, we hardly give references. We end with some historical notes.

Links to Wikipedia. This chapter contains a lot of terminology with which the average logician might not be familiar. We have used links to the relevant Wikipedia entries to facilitate the reader. When viewing this document in a PDF reader, clicking on the highlighted terms should open the relevant Wikipedia page in a browser.

1.1 Examples of decision problems in modal logic

This chapter is about solving problems in modal logic. What is a problem? A problem for us is a yes/no question. These problems are typically formalized as set-membership problems. Here are some examples.

Suppose first that a logic L is presented as a set of wffs (well formed formulas), as in Chapter 2 of this handbook. Then membership in L is the same as being valid. Thus the validity problem equals the L membership problem.

Alternatively, we can define a modal logic as a triple $(\mathsf{Wffs},\ \mathsf{Struc},\ \models)$, a set of wffs, a class of models, and a relation between the two. (To be precise, $\models$ is a relation between a model $\mathcal{M}$, a world w and a wff A.) This is essentially the way logics are defined in abstract model theory; see Chapter 1 of this handbook for further discussion.

In this richer setting, more natural decision problems show up:

Model checking

1. Given a finite model $\mathcal{M}$, is $\mathcal{M}$ a member of Struc?
2. Given a finite model $\mathcal{M}$ in Struc, a world w in $\mathcal{M}$ and a formula $A \in \mathsf{Wffs}$, does $\mathcal{M}, w \models A$ hold?

Satisfiability Given a formula $A \in \mathsf{Wffs}$, does there exists a model $\mathcal{M}$ in Struc and a world w in $\mathcal{M}$ such that $\mathcal{M}, w \models A$ hold?

Consequence Given a set of wffs C and a wff A, does $\mathcal{M} \models C$ implies $\mathcal{M} \models A$ for every model $\mathcal{M}$?

Model comparison problems Given two finite models, does there exists a bisimulation between the two?

Definability Given a set of wffs T and a propositional variable p, does T define p? The latter means that in any model of T, the interpretation of p is uniquely determined by the interpretation of the accessibility relations and the other propositional variables in T.

Proof Given a sequence s of wffs ending in A, is s a Hilbert style proof of A in some given axiom system?

Note that all these problems can be casted as set membership problems. There are also other types of problems whose complexity can studied, for example,

- Given that $A \to B$ is a validity, find a Craig interpolant C.
- Given a first order formula $\varphi(x)$ which is invariant for bisimulation, find the equivalent modal formula.

In the next subsection we have a closer look at the satisfiability problem, the model checking problem and the consequence problem. The wish to compute with models and wffs leads to certain desirable properties for logics. We collect three of these and derive some basic results for logics satisfying these properties. We finish this introduction with an example showing how different the satisfiability problem and the consequence problem may behave.

1.2 A simple and a hard problem

Consider a logic L presented as a set of wffs, as in Chapter 2 of this handbook. Then (1) is a natural problem.

(1) Given a string s, is s an element of L?

On close inspection (1) consists of two rather different problems:

(2) Given a string s, is s a wff?

(3) Given a wff s, is s an element of L?

Problem (1) is often called the validity problem, the set L being the set of valid wffs. Usually it is stated as (3). One can blur the distinction between (1) and (3) because logics are designed in a certain way. Namely it is assumed that problem (2) is much simpler than problem (3). In fact it is assumed that problem (2) can be solved in a practically feasible manner. Before we make this last notion precise let us look at an example. The wffs of the basic modal language were given by the grammar:

$$\varphi ::= p_i \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \Diamond\varphi.$$

We can see this as shorthand for the following recursive definition, given as a Prolog program:

```
wff(p_I)        :-   integer(I).
wff(¬F)         :-   wff(F).
wff((F ∧ G))    :-   wff(F), wff(G).
wff(◇F)         :-   wff(F).
```

The set of wffs computed by this program is the least fixed point of this recursive definition. In other words it is the smallest set of strings `s` such that `wff(s)` succeeds. Now suppose we run a computer with this program and ask it to determine whether `wff(s)` is true, for an arbitrary string `s`. Does it always terminate? And if so, can we estimate beforehand —based solely on the input `s`— when it will terminate? The first question is answered positively, and we can prove it by induction on the length of the input string. From this proof follows also a positive answer to the second question: The number of recursive calls is bounded by the number of symbols in `s`. This number is closely connected to the amount of time. For our purposes, it is not really relevant to measure time in seconds. For one thing, computers are getting faster every day, so that would make the results quickly outdated. But however fast a machine, it still has to do a number of "basic computation steps", given a program. Now, depending on the granularity of our analysis, we determine ourselves what a basic computation step is and what not. In the above example for instance, we may assume that for each string `s` it takes one computation step to match `s` with the argument of a head in one of the clauses. Then by induction on the length of `s` (notation $|\mathtt{s}|$), we can show that the machine needs at most $|\mathtt{s}|$ steps to find the answer (exactly $|\mathtt{s}|$ in case `s` happens to be a wff). We say that the time required by the machine is linear in the length of the input, or more informally we say that the program is in linear time. The notation which is often used is $O(|s|)$ (read this as "of the order $|s|$"). The big O notation is a way of stating bounds that ignores multiplicative constants and low order terms. This is an example of a computation which is practically feasible: the amount of time necessary grows proportionally with the length of the input.

Now we look at a program which is not practically feasible. Suppose you are given a propositional formula and the question is whether it is a tautology. This is a typical example of a task that one would like a computer to solve. From a theoretical point of view, the problem is clearly solvable. One can answer the question by writing out the truth table and checking whether the last column contains true in every row. This procedure just tries out all possible truth assignments to the propositional variables and checks whether each of them makes the formula true. The reader should convince him/herself that this check is easy: once a truth assignment is chosen, the check can be done using at most as many steps as the number of subformulas of the input formula. But unfortunately the number of checks grows exponentially with the number of propositional variables in the input. For a formula with n variables, 2^n checks have to be made. With some patience (on our side) a computer can carry this out for us. But each time the input formula contains just one more propositional variable, the time we may have to wait doubles. Very rapidly patience alone is not sufficient anymore: even on the fastest computers now available, we would need to wait longer than the lifetime of the universe for input strings containing as many variables as there are characters in this sentence. Thus the procedure is not practically feasible, as it takes time $2^{O(n)}$, for n the number of variables.

We have seen two examples, a linear and an exponential procedure. The practically feasible algorithms are commonly taken to be the procedures which on any input s terminate in at most $p(|s|)$ steps, where $p(x)$ is a polynomial function in x.

So, returning to the beginning, we wanted to solve problem (1) and showed that it splits into two subproblems (2) and (3). We really want to pay attention to problem (3). (In the case of propositional logic, we just saw that (3) seems much harder to solve than (2).) Now if a logic satisfies the following, we can safely ignore (2).

Desirable property 1. There exists a practically feasible algorithm which decides for any string whether it is a wff of the logic.

One can view the wffs as the *data* of the logician. In case of logics presented as a set of strings, they are the only data he has. The property states that access to the data is easy.

1.3 The model checking problem

As discussed above, a modal logic can also be defined as a triple (Wffs, Struc, $\models$), a set of wffs, a class of models, and a relation between the two. (To be precise, $\models$ is a relation between a model $\mathcal{M}$, a world w and a wff A.) Given these data, there is one obvious problem:

(4) Given a finite model $\mathcal{M}$ in Struc, a world w in $\mathcal{M}$, and a wff A, does $\mathcal{M}, w \models A$ hold?

This is often called the model checking problem. In Chapter 1, two procedures are presented which decide this problem for the basic modal language. It was assumed that wffs and models can be encoded as input to a computer program. But it was also assumed that the encodings were correct! That is, they really encoded models and wffs. But just as in the case of problem (1) deciding this is part of the problem. Let's see in detail how this works. The input of problem (4) consists of three parts

1. a string $s_{\mathcal{M}}$ encoding the model $\mathcal{M}$;
2. a string s_w encoding the world w;
3. a string s_A encoding the wff A.

To count as an admissible input to the problem (4) these strings have to satisfy certain properties:

(5) $s_{\mathcal{M}}$ encodes a model $\mathcal{M}$ and $\mathcal{M}$ is an element of Struc;

(6) s_w is the encoding of a world in $\mathcal{M}$;

(7) s_A is the encoding of a wff.

Problem (7) gave rise to our first desirable property. It sounds reasonable to ask the same for problem (5): thus we ask for a procedure which takes at most $p(|s_{\mathcal{M}}|)$ steps, for some polynomial function p to decide whether 1) the string encodes a model, and 2) whether the model belongs to Struc. The first part is not problematic; if we use reasonable encodings (cf. [1] or [10]) this step is practically feasible. The second part, checking whether the model belongs to Struc, is quite a different matter. Here we do not check a simple property, like whether a string is a wff, or encodes a model. Here we check whether the model belongs to a class of models. This is a problem for which it is not even clear that it is decidable in general. In fact [4] (exercise 6.2.4) asks the reader to create a *logic* for which this problem is undecidable (a simple cardinality argument suffices to show that there must be classes for which membership is undecidable).

In modal logic it is common to define the class of models Struc as the class on which a finite set of modal formulas is valid. That means —if the truth definition is first-order— that the class is defined by a sentence from monadic second-order logic of the following shape:

> block of universal quantifiers ranging over sets followed by a first order formula in the signature of the accessibility relations of the logic.

Intuitively, the language in which we define the class of models influences the complexity of deciding membership in that class: the greater expressivity the language offers, the more difficult it

can be to decide membership. This intuition is made precise in the field of descriptive complexity theory [23]. Starting with Fagin's Theorem [12] it turned out that classes of models which are defined by their complexity in deciding membership correspond exactly to classes defined in certain well-known logical formalisms. Fagin's Theorem states that the following are equivalent, for C a class of finite structures,

- membership in C can be decided in non deterministic polynomial time in the size of the input structure;
- C is definable by a sentence of existential second-order logic.

Problems which can be decided in non-deterministic polynomial time are said to be in NP. Boolean satisfiability is the most prominent example of a problem in NP. As stated above some NP problems are generally believed not to be practically feasible. For more about NP, see Section 3. A class of structures defined by a universal second-order sentence is the complement of a class of structures defined by an existential second-order sentence. The complexity class co-NP consists of all problems whose complement is in the complexity class NP. (In computational complexity theory, the complement of a decision problem is the decision problem resulting from reversing the yes and no answers.) Thus Fagin's Theorem also connects the complexity class Co-NP with universal second-order definability. It is also generally believed that Co-NP hard problems cannot be solved in a practically feasible manner.

Desirable property 2.

- The class of models of a logical system is definable by a universal second-order sentence.
- Equivalently, the class is defined such that checking membership for finite models is in co-NP.

The most natural way of defining classes of structures in modal logic —by giving a number of modal axioms— leads in general to an impractical decision procedure. But of course this is the general case. For instance, it is easily seen that checking whether a model satisfies the axiom $\Diamond\top$ is practically feasible. In fact, it turns out that most well-known modal logics have a practically feasible model recognition problem, for instance all logics axiomatized by Sahlqvist axioms. This is because for first-order definable classes the membership problem is decidable in polynomial time. In fact this even holds for a powerful second-order extension of it [23]:

THEOREM 1 (Immerman–Vardi). *Let C be a class of finite structures. If C is defined by a sentence in first-order logic expanded with a least fixed point operator (in notation FO(LFP)), then checking membership in C can be done in polynomial time in the size of the input structure.*

Compared to Fagin's Theorem, Theorem 1 only states one (the easy) direction, and not an equivalence. In fact it is still an open problem to find the language corresponding to polynomial time. (Though it is solved on *ordered* structures: i.e., models which come with a linear order on their domain. Restricted to classes of *ordered* structures the converse of Theorem 1 holds as well, which is the full Immerman-Vardi Theorem.)

In the rest of this chapter we will only deal with modal systems whose class of models is definable in FO(LFP), which implies that membership is decidable in polynomial time.

The complexity of $\models$.

Having settled the hairy details, we can finally look at the problem (4). What we really ask here is the complexity of the relation $\models$. As models and formulas are the data of logic we must be able to access the data, thus we insist that (4) is decidable. How difficult it is to decide (4) in a particular case is a topic carefully studied in finite model theory and in database theory [1, 10]. Note that in the formulation of (4) the complexity is solely due to the expressive power of the language. There is something particular about the design of many modal languages which cause (4) to be practically feasible. We explain that now. As indicated in Chapter 1, the definition of $\models$ is given in terms of first-order logic. As we can see from the standard translation, the basic modal language corresponds to a fragment of first-order logic which contains only two variables. Obviously this is also true when considering more than one modality, but fails when considering polyoadic modalities; see Chapter 5, Section 1.5. But still the meaning of an n-ary modality is defined there using $n+1$ variables. So each modal similarity type τ which has a bound n on the arity of its operators corresponds to a $n+1$ variable fragment of first-order logic. Note that the definition of modal similarity type is general enough to allow a type with infinitely many modalities, each having a different arity. Such modal languages do not correspond to finite variable fragments. Being a bounded variable fragment seems to be a distinguishing feature of modal languages. There is also a nice complexity theoretic argument for requiring boundedness in terms of variables: the model checking problem for each bounded variable fragment of first order logic is practically feasible, while it is not for full first order logic [38].

THEOREM 2 (Immerman–Vardi). *Given a first-order formula $\varphi(\bar{x})$ in a fixed bounded variable fragment, a first-order model $\mathcal{M}$ and a sequence $\bar{a}$ of elements from the domain of $\mathcal{M}$, it is decidable in polynomial time in $|\varphi(\bar{x})|$ and $|\mathcal{M}|$ whether $\mathcal{M} \models \varphi(\bar{a})$.*

This gives us the third desirable property: the definition of $\models$ must be given in terms of a bounded variable fragment of first-order logic. Note that $\models$ could alternatively be defined as the standard translation.

Desirable property 3. The definition of $\models$ is given as a polynomial time computable function from the set of wffs to a bounded variable fragment of first order logic.

We immediately obtain that in favorable cases the model checking problem is practically feasible.

THEOREM 3. *Let L be a logical system satisfying the desirable properties 1 and 3. Let its class of models be defined by a FO(LFP) sentence. Then the model checking problem is decidable in polynomial time.*

1.4 The consequence problem

The consequence problem $C \models A$ is the central problem in logic. Here we restrict to finite C and study its complement:

(8) Given wffs C and A, can A be satisfied on a model $\mathcal{M} \in \mathsf{Struc}$ which globally satisfies C?

Recall that $\mathcal{M}$ globally satisfies C if $\mathcal{M}, w \models C$ holds for *all* worlds w in $\mathcal{M}$. We call this problem *satisfiability under constraints*. Without C we call it simply the *satisfiability* or *local satisfiability* problem.

There is no reason to believe a priori that this problem is decidable for a given modal system, even if it satisfies all design criteria. However there exists a sufficient condition which yields decidability in a straightforward manner.

DEFINITION 4. Let f be some computable function. A logical system $(\mathsf{Wff}, \mathsf{Struc}, \models)$ is said to have the f–bounded model property if for all wffs C, A, it holds that whenever A is satisfiable in a C model (in Struc) it is satisfiable in a C model (in Struc) whose size is bounded by $f(|C|, |A|)$.

THEOREM 5. *For any logical system having all three desirable properties, and having the f–bounded model property, problem (8) is decidable.*

Proof. Consider arbitrary wffs C and A. Then by the bounded model property the answer to (8) is yes if and only if there exists a model $\mathcal{M}$ in Struc whose size is bounded by $f(|C|, |A|)$ which globally satisfies C and locally satisfies A. Up to isomorphism there are finitely many structures of that size, so we can write a procedure which lists all of them (using some representation). By property 2 we can decide whether a structure is in Struc. By Theorem 2 we can check in polynomial time in $f(|C|, |A|)$, $|C|$ and $|A|$ whether a structure globally satisfies C and locally satisfies A. So we have to make a finite number of such checks, which yields the theorem. ❑

The procedure sketched in the last proof highlights a particular feature of algorithms designed to solve a problem in which one asks for the existence of a certain object. Note that this is the distinguishing feature between problems (4) and (8). In problem (4) we are given three objects, a model, a world and a wff and we have to determine whether they stand in the $\models$ relation. In problem (8) we are just given the constraint C and the wff A and we asked for the existence of a model. The algorithm for problem (8) uses the algorithm for problems (4) and (5) every time it “tries” a model from its long list of candidates. So the algorithm for (8) naturally divides into two parts:

- a (difficult) *search* for the right candidate model;
- an (easy) *check* that the candidate is correct (i.e., the model is in Struc, it globally satisfies C, and locally satisfies A).

For these kind of problems a special computational model has been designed which reflects the sketched division of labor. These are called non-deterministic computations. For a non-deterministic computation, the search for the right candidate takes no more time than required to write it down plus the time it takes to check whether a potential candidate is a real candidate. This discussion is captured in the next theorem:

THEOREM 6. *Let the logical system* $(\mathsf{Wff}, \mathsf{Struc}, \models)$ *satisfy all desirable properties. If* Struc *is defined by an FO(LFP) sentence and the system has the f–bounded model property, then there is some polynomial p such that (8) is decidable by a non-deterministic algorithm taking time* $p(f(|C|, |A|), |C|, |A|)$.

The proof of Theorem 18 in Chapter 1 can be extended to show that the basic modal logic has the $2^{(|C|+|A|)}$ bounded model property; see Proposition 10 below. Thus (8) can be decided by a non-deterministic algorithm which takes exponential time in the length of the input. Later on we see that we can do better, the non-determinism is not necessary. We finish the introduction by giving an example of a logic for which (8) is undecidable in general but decidable in non-deterministic polynomial time in the special case when the set of constraints is empty.

1.5 A tiling logic

We present an undecidable problem which is particularly well-suited for modal logics. Moreover finite variants of it exist which are useful in proving complexity results as we will see later on. These are the *tiling problems*. A tile is a one-by-one square which has a 'color' on each of its sides; these colors are given by four functions 'right', 'left', 'up' and 'down'. Given a set T of tiles containing one special tile T_0, a *tiling* of the grid $\mathbb{N} \times \mathbb{N}$ by T is a map t from $\mathbb{N} \times \mathbb{N}$ to T satisfying, for all $n, m \in \mathbb{N}$:

$$\begin{array}{lcl} t(0,0) & = & T_0, \\ \text{right}(t(n,m)) & = & \text{left}(t(n+1,m)), \\ \text{up}(t(n,m)) & = & \text{down}(t(n,m+1)). \end{array}$$

Tiles are assumed to be fixed in orientation, so the above conditions say that colors of adjacent tiles match. (We note that it is not required to use all tiles of T in a tiling of $\mathbb{N} \times \mathbb{N}$.) If such a tiling exist, we say that T *can tile* $\mathbb{N} \times \mathbb{N}$.

The following problem is undecidable.

$\mathbb{N} \times \mathbb{N}$ **tiling**: Given a finite set T of tiles, can T tile $\mathbb{N} \times \mathbb{N}$?

We will now define a modal system Tile which is tailored to encode the above tiling problem. The language of Tile contains two unary modalities $\langle \mathsf{right} \rangle$ and $\langle \mathsf{up} \rangle$. In a model of the form (W, R_r, R_u, V), these modalities receive their meaning in the usual way:

$$\begin{array}{lcl} \mathcal{M}, s \models \langle \mathsf{right} \rangle \varphi & \iff & \mathcal{M}, t \models \varphi \text{ for some } t \text{ with } R_r st, \\ \mathcal{M}, s \models \langle \mathsf{up} \rangle \varphi & \iff & \mathcal{M}, t \models \varphi \text{ for some } t \text{ with } R_u st. \end{array}$$

The class Struc of models of Tile consists of all models (W, R_r, R_u, V) in which R_r and R_u are (the graphs of) two commuting total functions. In particular, grid models satisfy the following condition:

(9) $\forall xyz((R_r xy \wedge R_u xz) \rightarrow \exists w(R_r zw \wedge R_u yw))$.

THEOREM 7. *Problem (8) is undecidable for the logic* Tile.

Proof. We reduce the $\mathbb{N} \times \mathbb{N}$–tiling problem to the satisfiability problem for Tile. We present a procedure that outputs for every instance T of the tiling problem, wffs C_T and A_T such that the following are equivalent

- T can tile $\mathbb{N} \times \mathbb{N}$;
- there exists a model $\mathcal{M}$ in Struc which globally satisfies C_T and locally satisfies A_T.

Take for any set $T = \{T_0, T_1, \ldots, T_k\}$ of tiles a corresponding set $\{t_0, t_1, \ldots, t_k\}$ of propositional variables. Let A_T be t_0. Define C_T as the conjunction of the following formulas (where i ranges over $0,\ldots,k$):

$$\begin{array}{lll} A1 & \bigvee_{0 \le i \le k} t_i & \\ A2 & \bigwedge_{0 \le i \le k} & \left[t_i \rightarrow \bigwedge_{i \ne j} \neg t_j \right] \\ A3 & \bigwedge_{0 \le i \le k} & [t_i \rightarrow \langle \mathsf{right} \rangle \bigvee \{ t_j \mid \text{right}(T_i) = \text{left}(T_j) \}] \\ A4 & \bigwedge_{0 \le i \le k} & [t_i \rightarrow \langle \mathsf{up} \rangle \bigvee \{ t_j \mid \text{up}(T_i) = \text{down}(T_j) \}] \,. \end{array}$$

It is almost immediate that T tiles $\mathbb{N} \times \mathbb{N}$ if and only if there exists a Tile model where C_T holds throughout and t_0 is satisfied at some world. (The reader should verify that in hard direction of the proof property (9) of grid models is crucial.) Thus we have reduced the undecidable tiling problem to problem (8) for the logic Tile. ❑

We hasten to remark that the undecidability of this system has nothing to do with the fact that we are dealing with more than one modality here; one can easily transform this example into an undecidable modal system in the *basic modal language*.

It is interesting to note that *without* constraints which hold globally these grid logics become quite harmless. In fact, the grid–like nature of their models ensures that every locally satisfiable formula A is satisfiable in a model whose size is at most $(|A|+1)^2+1$.

THEOREM 8. *Every locally* Tile *satisfiable formula A is satisfiable in a* Tile *model of size at most* $(|A|+1)^2+1$. *As a corollary,* Tile *has a local satisfiability problem which is decidable in non-deterministic polynomial time.*

Proof. Let $\mathcal{M}$ satisfy A at s. Let k be the modal depth of A. Thus $k \leq |A|$. Let $\mathcal{M}'$ be the smallest substructure of $\mathcal{M}$ which contains s together with all states reachable in at most k (R_r- or R_u-) steps from s and which satisfies property 9. By Lemma 21 in Chapter 1, A is still satisfied in the model $\mathcal{M}'$. The size of the universe of $\mathcal{M}'$ is at most $(k+1)^2$. Unfortunately $\mathcal{M}'$ is not yet a Tile model, because not every state has a R_r and R_u successor. In order to mend this, add one dummy state x to the universe of $\mathcal{M}'$ and put a link from w to x for all states w (including x itself) that do not have a successor yet. That is, define $W^+ = W' \cup \{x\}$, $R_r^+ = R_r' \cup \{(w,x) \mid R_r'wy$ for no y in $\mathcal{M}'\}$, and likewise for R_u^+. Let the valuation stay the same, i.e., define $V^+(p) = V'(p)$ for all p.

The resulting model $\mathcal{M}^+$ is a Tile model. Clearly A is still satisfied at s in this new model, since x is 'too far away' to have any effect on the truth of A. (The k-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$ is also a k-bisimulation between $\mathcal{M}$ and $\mathcal{M}^+$.) This proves the first part of the theorem. The complexity result follows from Theorem 6. ❑

2 DECISION ALGORITHMS

We have seen some very general results and some very basic desirable properties. Now we look at actual algorithms for deciding the satisfiability problem under constraints and the local satisfiability problem for a number of typical cases.

All presented decision algorithms are based on the following idea: show that for each wff A, the following are equivalent:

(10) A is satisfiable on a Struc model.

(11) There exists a finite structure $\mathcal{M}_A$ satisfying

- a finite number of decidable properties, and
- the size of $\mathcal{M}_A$ is bounded by some function $f(|A|)$.

From the discussion in the previous section it should be clear that decidability of the local satisfiability and the validity problems follow from this. A similar reduction will be given for the satisfiability problem under constraints. Note that an upper bound on the time used by the algorithm follows from 1) the function f and 2) the difficulty of checking the properties on $\mathcal{M}_A$.

First let's look at the kind of structures $\mathcal{M}_A$ we are after and what kind of properties we can expect. In the simplest case we just ask that $\mathcal{M}_A$ belongs to Struc and $\mathcal{M}_A, w \models A$ for some world w. Often this is either not possible (for instance, if the logic does not have the finite model property) or $\mathcal{M}_A$ gets unnecessarily large.

For the first case, consider the basic modal language interpreted on models $(\omega, \mathsf{succ}, V)$, hence

$$n \models \Diamond A \iff n+1 \models A.$$

Obviously this logic does not have the finite model property as Struc consists of infinite models only. Still using the technique from the proof of Theorem 8 we can show that each satisfiable A is satisfiable at world 0 in a model $(\{0, 1, \ldots, \mathrm{modal\_depth}(A), \mathsf{dummy}\}, \mathsf{succ}', V)$, which is linear in $|A|$. Here succ' is the set $\{(n, n+1) \mid n < \mathrm{modal\_depth}(A)\} \cup \{(\mathrm{modal\_depth}(A), \mathsf{dummy}), (\mathsf{dummy}, \mathsf{dummy})\}$.

For the second case consider the same language but interpreted on finite binary trees. Let $\Diamond$ be interpreted by the first_child relation. By the same reasoning as above any satisfiable A can be satisfied at the root of a tree of depth $\mathrm{modal\_depth}(A)$. But a binary tree of this depth contains $2^{\mathrm{modal\_depth}(A)}$ many leaves! This model contains a lot of useless information. The only important part for the satisfiability of A at the root is its left most branch.

So in both cases we could do with a *pseudo-model* and that is what $\mathcal{M}_A$ is in general. The relation with some real model can be the identity, it can be a $\mathrm{modal\_depth}(A)$ bounded bisimulation or a more intricate or ad hoc relationship. In principle everything is fine as long as the equivalence between (10) and (11) holds.

We will now review the most popular techniques to create structures $\mathcal{M}_A$. We interleave the analysis with examples of formulas enforcing large models.

2.1 Selection of points

Let A be a satisfiable basic modal wff. Then, by Theorem 22 in Chapter 1, A is satisfiable at the root of a tree of depth at most $\mathrm{modal\_depth}(A)$. This looks like a good candidate for a finite model, the only problem is that it can be infinitely branching. Here selection of points comes in. We first need an important concept: the set of A relevant formulas.

All relevant formulas will be subformulas of A; we need a bit extra though. Given a formula B, let $\sim B$ denote the formula C if B is of the form $\neg C$; otherwise, $\sim B$ is the formula $\neg B$; we say that a set Σ of formulas is closed under taking single negations if $\sim B \in \Sigma$ whenever $B \in \Sigma$. This notion enables us to pretend that a finite set is closed under taking negations by treating $\sim B$ as if it were the real negation of B. Now given a set of formulas Σ, let $Cl(\Sigma)$ be the smallest set of formulas that extends Σ and is closed under taking subformulas and single negations. When A is a formula, we denote the set $Cl(\{A\})$ of relevant A formulas by $Cl(A)$; it is easy to see that the cardinality of $Cl(A)$ is linear in the length of A. We call $Cl(A)$ the *closure* of A.

Now let $\mathcal{M}$ be the chopped tree of finite depth which satisfies A at the root. For any world w in $\mathcal{M}$ define the A theory of w (notation: $\theta^A_{\mathcal{M}}(w)$) as the set $\{B \in Cl(A) \mid \mathcal{M}, w \models B\}$. Now if we can create a model $\mathcal{M}_A \subseteq \mathcal{M}$ such that

root the root of $\mathcal{M}$ is still in $\mathcal{M}_A$, and

succ for each w in $\mathcal{M}_A$, $\theta^A_{\mathcal{M}}(w) = \theta^A_{\mathcal{M}_A}(w)$,

then A is still satisfied at the root of $\mathcal{M}_A$. We create $\mathcal{M}_A$ by selecting for each world w just enough successors in order to ensure that $\theta^A_{\mathcal{M}}(w) = \theta^A_{\mathcal{M}_A}(w)$. Starting at the root, choose, for every subformula of A of the form $\Diamond\varphi$, a successor of the root at which φ is true (if such a successor exists at all). Obviously, at most b successors need to be chosen, where b is the number of diamond subformulas of A. Hence, by deleting all successors that were not chosen and their descendants from the model, we obtain a tree model whose branching degree at the root is at

most b. A simple verification shows that A still holds at the root. Now repeat this process at each of the chosen successors of the root and continue until the leaves of the tree are reached. Obviously A is still satisfied at the root. The result is our desired $\mathcal{M}_A$. Clearly conditions **root** and **succ** are satisfied and we have shown

PROPOSITION 9. *Any satisfiable modal formula A can be satisfied at a the root of a finite tree model with the following properties:*

- *the depth is bounded by the modal depth of A, and*
- *the branching degree is bounded by the number of diamond subformulas of A.*

The number of worlds in this model is exponential in the modal depth of the formula A. It seems that in the worst case such a size is unavoidable. We now show a formula which exemplifies this worst case behavior. We define, for each natural number n, a satisfiable formula $A(n)$ with the following two properties

- the size of $A(n)$ is quadratic in n, and
- when $A(n)$ is satisfied in any model $\mathcal{M}$ at state s, then $\mathcal{M}$ contains as a substructure an isomorphic copy of the binary tree of depth n whose root is s.

Thus the size of the smallest model satisfying $A(n)$ is exponential in $|A(n)|$. The idea underlying the definition of $A(n)$ is very simple: take n propositional variables $p_0, \ldots, p_{n-1}$ and write a formula which when satisfied forces a binary branching tree in which every possible valuation on $\{p_0, \ldots, p_{n-1}\}$ occurs at some leaf. Thus the model certainly contains 2^n different states. The formula is constructed using two macros: $\mathit{branch}(p_i)$ and $\mathit{store}(p_i)$ defined as follows:

$$\begin{array}{lcl} \mathit{branch}(p_i) & := & \Diamond p_i \wedge \Diamond \neg p_i \\ \mathit{store}(p_i) & := & (p_i \rightarrow \Box p_i) \wedge (\neg p_i \rightarrow \Box \neg p_i). \end{array}$$

The formula $A(n)$ then is given as

$$(12)\quad \mathit{branch}(p_0) \wedge \bigwedge_{1 \leq i < n} \Box^i \left[\mathit{branch}(p_i) \wedge \bigwedge_{0 \leq j < i} \mathit{store}(p_j) \right],$$

in which $\Box^i$ abbreviates a sequence of i many boxes. The formula works as follows. Suppose $\mathcal{M}, s \models A(n)$. Then the branch part of $A(n)$ states that every node t reachable in i R–steps from s has two different successors, one forcing p_i and another forcing $\neg p_i$. The store part of the formula states that successors of t created by the branch part satisfy precisely the same proposition letters $p_0, \ldots, p_{i-1}$ as does t. We leave it to the reader to verify that the interplay of the branch and store macros forces a binary tree of depth n, as desired.

2.2 Filtration

Let C, A be wffs. Assume that there exists a model $\mathcal{M}$ and a world w_0 such that $\mathcal{M} \models C$ and $\mathcal{M}, w_0 \models A$. Now if C contains for instance the formula $\Diamond\top$, then $\mathcal{M}$ cannot be a finite tree. So the selection of points method does not work in the presence of global constraints. Instead we will choose enough worlds from the model $\mathcal{M}$ to create a new small model.

We will show that there exists a model $\mathcal{M}'$ containing $2^{O(|C|+|A|)}$ many worlds such that $\mathcal{M}' \models C$ and $\mathcal{M}', w \models A$. The model $\mathcal{M}'$ is defined from the theories $\theta^{A,C}_{\mathcal{M}}(w)$, for worlds w in $\mathcal{M}$. Thus the domain of $\mathcal{M}'$ is the set

$$\{\theta^{A,C}_{\mathcal{M}}(w) \mid w \text{ in } \mathcal{M}\}.$$

Note that the worlds in $\mathcal{M}'$ are sets of formulas. We want to create a model such that each world describes which relevant wffs are true in it. I.e., we want a

Truth Lemma For all wffs $B \in Cl(A, C)$, for all worlds $\theta^{A,C}_{\mathcal{M}}(w)$,

$$B \in \theta^{A,C}_{\mathcal{M}}(w) \text{ if and only if } \mathcal{M}', \theta^{A,C}_{\mathcal{M}}(w) \models B.$$

The desire to have this truth lemma puts three constraints on the model $\mathcal{M}'$:

valuation V' is defined such that for all propositional variables p_i,

$$p_i \in \theta^{A,C}_{\mathcal{M}}(w) \text{ if and only if } \theta^{A,C}_{\mathcal{M}}(w) \in V'(p_i).$$

Relation R' is defined such that

- **min** For all worlds x, if $\Diamond B \in x$, then there exists a world y such that $R'(x, y)$ and $B \in y$.
- **max** For all worlds x, y in $\mathcal{M}'$, for all $\Diamond B \in Cl(A, C)$ if $R'(x, y)$ then $B \in y$ only if $\Diamond B \in x$.

The truth lemma can be shown by induction on the length of the wff. It holds for every model $\mathcal{M}'$ satisfying these three constraints. The first constraint determines the definition of V'. For R', we have more freedom. Any relation satisfying both **min** and **max** will do. For instance, R' can be defined as follows: $R'(x, y)$ if and only if for all $\Diamond B \in Cl(A, C)$, $B \in y$ only if $\Diamond B \in x$. Note that when proving the truth lemma we use two properties of the sets $\theta^{A,C}_{\mathcal{M}}(w)$ which together state that these sets of formulas do not contain blatant propositional inconsistencies:

and For $B \wedge D \in Cl(A, C)$, $B \wedge D \in \theta^{A,C}_{\mathcal{M}}(w) \iff B \in \theta^{A,C}_{\mathcal{M}}(w)$ and $D \in \theta^{A,C}_{\mathcal{M}}(w)$.

not For $B \in Cl(A, C)$, $\sim B \in \theta^{A,C}_{\mathcal{M}}(w) \iff B \notin \theta^{A,C}_{\mathcal{M}}(w)$.

The model $\mathcal{M}'$ thus obtained is called the *filtration* of the model $\mathcal{M}$ through the set of formulas $Cl(A, C)$. Note that —by the truth lemma— $\mathcal{M}' \models C$, and —because $\theta^{A,C}_{\mathcal{M}}(w_0)$ is a world in $\mathcal{M}'$— $\mathcal{M}', \theta^{A,C}_{\mathcal{M}}(w_0) \models A$. Thus we have shown

PROPOSITION 10. *If A is satisfiable in a model which globally satisfies C, then A is satisfiable in a model which globally satisfies C and whose set of worlds is bounded by $2^{O(|C|+|A|)}$.*

2.3 Hintikka set elimination

We now give an algorithm which *constructs* the model whose existence was just shown. The idea comes straight from the proof of the truth lemma. That inductive proof shows that if we can find a set G of subsets of $Cl(A, C)$ such that

HS1 every element of G contains C;

HS2 there is an element of G which contains A;

HS3 every element of G satisfies the properties **and** and **not**

HS4 for every element X of G, for every formula $\Diamond B \in Cl(A, C)$, if $\Diamond B \in X$, then there exists a $Y \in G$ such that

1. $B \in Y$, and
2. for all $\Diamond D \in Cl(A, C)$ if $D \in Y$ then $\Diamond D \in X$.

then, we can prove the truth lemma, and A is satisfiable in a model which globally satisfies C. The filtration described above shows the other direction: if A is satisfiable in a model which globally satisfies C, then a set G satisfying HS1–HS4 exists. So we describe a procedure which tries to create that set G.

Let S_0 consists of all sets $\Delta \subseteq Cl(A, C)$ which contain C and which satisfy properties **and** and **not**. Thus conditions HS1 and HS3 hold for S_0. Clearly S_0 can be effectively computed and $|S_0| \leq 2^{O(|A|+|C|)}$. We now inductively construct a sequence of sets of sets of formulas $S_0 \supsetneq S_1 \supsetneq S_2 \supsetneq S_3 \cdots$. During this construction we try to find witnesses for diamond formulas. We say that a set $X \in S_i$ is *ready* if only for the set X, condition HS4 holds with G replaced by S_i. In other words, the set S_i contains witnesses for all diamond formulas in X. If every set in S_i is ready and S_i satisfies HS2, then return 'A is satisfiable in a global C model'. If there is no set in S_i containing A, then return the negation of the last statement. Otherwise, let S_{i+1} consists of all ready sets in S_i and continue the construction. Since $S_i \supsetneq S_{i+1}$, the construction is guaranteed to terminate in at most $2^{O(|A|+|C|)}$ stages.

Why is this algorithm correct? If the algorithm answers 'satisfiable', then it has found a set of subsets of $Cl(A, C)$ satisfying the four HS conditions. But then we can create a model out of them, just as we did in the filtration. For instance, we can define R to be minimal. Now we use the conditions to show that the truth lemma holds. Conversely, suppose that A is locally satisfiable in a C model $\mathcal{M}$. Let G be the set $\{\theta^{A,C}_{\mathcal{M}}(w) \mid w \text{ in } \mathcal{M}\}$. It is easy to show that G satisfies the four HS conditions, and that the algorithm will never delete any element in G in any of its stages. Thus it will return 'satisfiable'.

How many computation steps does this procedure take? It lasts at most $2^{O(|A|+|C|)}$ stages. At every stage we check whether S_i satisfies properties HS2 and HS4. How can we find out how long each check takes? The simplest way is to formalize the whole procedure in terms of first-order logic and use the results about first-order model checking. We can view the power set of $Cl(A, C)$ as the domain of a first order model, which has unary predicates for each formula in $Cl(A, C)$. Then the conditions HS1–HS4 become just first-order conditions on that model. For instance, we have

$$\forall x(P_{\sim B}(x) \leftrightarrow \neg P_B(x)).$$

with P_B the unary predicate corresponding to the modal formula B. We have to write as many of such first-order conditions as there are subformulas of A and C. Also note that they can all be written using just two first-order variables. But we know that given a model and a first-order formula in a fixed number of variables, checking whether that formula is true in that model can be done in time polynomial in the size of the model and the formula. But the size of each model corresponding to a set S_i is bounded by $2^{O(|A|+|C|)}$. So each check can be done in a polynomial number of steps in $2^{O(|A|+|C|)}$, which is just $2^{O(|A|+|C|)}$ many steps. We must make at most $2^{O(|A|+|C|)}$ many such checks, so the whole algorithm takes $2^{O(|A|+|C|)}$ many steps.

Subsets of $Cl(A, C)$ satisfying HS3 are called Hintikka Sets after Jaakko Hintikka who first employed them.

2.4 Hintikka set elimination without constraints

The last algorithm is very impractical: it takes exponentially many steps for every input. This is because in the first step it already creates all possible candidates for worlds in the model. When the set of constraints is empty we can do better using a non-deterministic algorithm. The idea is the following. Let A be the formula for which we want to decide whether it is locally satisfiable. Guess a set $X \subseteq Cl(A)$ and check whether it satisfies HS2 and HS3. Suppose it does. Then it is a candidate for the root of a model for A. Now we must check HS4 for X. Instead of finding the witnesses Y in some set of candidates, we create them on the fly. Now the important point is what the structure of these Y should be. As the Y will be placed in a model one step away from the root, it is wasteful to prove the truth lemma for Y for all formulas in $Cl(A)$. In order to prove the truth lemma at the root X, we only need that it holds at Y for the set of formulas $Cl\{B|\Diamond B \in Cl(A)\}$. And so forth for every next level. (We urge the reader to verify this.) This is the idea behind the next algorithm: the further we are away from the root, the less diamond formulas we have to find witnesses for. In fact, the recursion depth of the algorithm is bounded by exactly the modal depth of the input formula A. What the algorithm is really doing is searching for the tree model that we constructed in the selection of points proof given above.

The algorithm presented in Figure 1 implements this search for a tree model. We claim that for sets of formulas Δ and Σ such that Σ is closed under taking subformulas and single negations, *K–World*(Δ, Σ) will be true iff there exists a tree model $\mathcal{M}$ such that at the root s, for all $B \in \Sigma$, $(\mathcal{M}, s \models B \iff B \in \Delta)$. This function can be used to solve satisfiability for the basic modal system, since A is satisfiable iff there exists a set $\Delta \subseteq Cl(A)$ such that $A \in \Delta$ and *K–World*$(\Delta, Cl(A))$ is true.

Note that with each recursive call of *K-World*, the size of the set Σ decreases, since we include formulas of smaller modal depth only. Thus the recursion depth is bounded by the modal depth of the input formula A. That the function is correct can be shown by induction on the size of Σ; we leave this to the reader. Now what about the complexity of this algorithm? Obviously, if we feed it the binary-branching tree formula given below Proposition 9 it needs as many steps as there are nodes in the tree, 2^n, for n the number of propositional variables in the formula. But besides the number of steps, we also measure the amount of memory space a machine needs for a specific algorithm. In the Hintikka set elimination algorithm the space used was exponential in the input formula, as the set of all candidates had to be stored. At first sight, the K–World algorithm needs just as much space, as it will create a complete tree model, if it can. But K–World can be implemented in such a way that it only needs to store at most one complete branch of the tree model plus a little storage for administration. The idea is that once it has checked that it can create certain desired witnesses, it can remove those witnesses from memory. It just needs to remember not to do that check again, which is easily implemented. As the length of each branch is bounded by the modal depth of A, the whole algorithm needs a polynomial amount in $|A|$ of memory space. This argument is made precise in Chapter 4.

2.5 Forcing exponentially deep paths

Now we will see that global constraints destroy the polynomially bounded depth of paths of the satisfying models for the basic modal system. In particular, we create a globally satisfiable constraint which, when satisfied, forces a branch in the model containing an exponential number of different Hintikka sets. The algorithm sketched in the previous subsection used only polynomial space because the paths in the satisfying model could be kept “short”.

Assume that Δ and Σ are finite sets of formulas such that $\Delta \subseteq \Sigma$ and Σ is closed under taking subformulas and single negations.

K–World(Δ, Σ) if and only if

- Δ satisfies properties **and** and **not** with respect to Σ, that is,

  not For $B \in \Sigma$, either $\sim B \in \Delta$ or $B \in \Delta$.

  and For $B \wedge D \in \Sigma$, $B \wedge D \in \Delta \iff B \in \Delta$ and $D \in \Delta$.

- for each formula $\Diamond B \in \Delta$ there is a set $\Delta_B \subseteq \Sigma$ such that
  - $B \in \Delta_B$,
  - $(\forall \Diamond D \in \Sigma) : D \in \Delta_B \Rightarrow \Diamond D \in \Delta$, and
  - *K–World*$(\Delta_B, \mathit{Cl}(\{D \mid \Diamond D \in \Sigma\}))$.

Figure 1. The function *K-World* decides **K** satisfiability.

A simple way of forcing the existence of exponentially deep R-paths is to employ binary counters. By a binary counter we will understand a device that can have natural numbers as values, represented as binary strings of 0s and 1s; one should also be able to increment this value by one. We will use a set $\{p_0, \ldots, p_{n-1}\}$ of propositional variables to implement an n-ary binary counter (n-ary means that the counter is reset to zero after reaching $2^n - 1$). We use these variables to encode the n bits of the counter, with p_0 encoding the least significant and p_{n-1} the most significant bit. The variable p_i being true in a given state, encodes the fact that the ith bit of the counter is 1 in that state. The key idea to an encoding into the modal language lies in the following characterization of adding 1 to a binary counter. If $a = a_{n-1} \ldots a_0$ and $b = b_{n-1} \ldots b_0$ are two n–bit counters, then $b = a + 1 \pmod{2^n}$ precisely when the following holds: either $b_i = 0$ and $a_i = 1$ for all i (this is when we start counting at 0 again), or, for some $k \leq n - 1$ we have that

(1) $a_k = 0$, and $b_k = 1$,

(2) $a_j = 1$ and $b_j = 0$ for all $j < k$, and

(3) $a_i = b_i$ for all $i > k$.

In a picture:

$$\begin{array}{llll} 10110 & 0 & 1111 & \mathbf{a} \\ 00000 & 0 & 0001 & \\ \hline 10110 & 1 & 0000 & \mathbf{b = a + 1}. \\ & \mathbf{k} & & \end{array}$$

We want to write wffs $A(n)$ and $C(n)$ which force a counter to take on all values from 0 to $2^n - 1$, in consecutive states. In particular, we want that if $\mathcal{M} \models C(n)$ and $\mathcal{M}, s_0 \models A(n)$, then $\mathcal{M}$ contains an R-path of length $2^n - 1$ starting at s_0. Moreover all sets $\theta_{\mathcal{M}}^{\{p_0,\ldots,p_{n-1}\}}(s_i)$, for s_i lying on this path, are different. We take care that the formulas have length only $O(n^2)$. The formula $A(n)$expresses the fact that the counter is initially set to 0:

$$\neg p_0 \wedge \ldots \wedge \neg p_{n-1}.$$

The formula $C(n)$ will be a conjunction of three wffs which should hold globally in a model. The first conjunct expresses that every state has a successor:

$$\Diamond\top.$$

The next two conjuncts take care of addition. They express that whenever an R–transition is made in the model the binary counter is increased by one. First the simple case of resetting the counter:

$$(p_0 \wedge \ldots \wedge p_{n-1}) \rightarrow \Box(\neg p_0 \wedge \ldots \wedge \neg p_{n-1}).$$

Finally, the last conjunct of $C(n)$ covers the case when we have to 'carry one'. This conjunct will itself be a conjunction, having a conjunct of the following form for every k such that $0 \leq k < n$:

$$(\neg p_k \wedge \bigwedge_{j<k} p_j) \rightarrow \Box(p_k \wedge \bigwedge_{j<k} \neg p_j) \wedge \bigwedge_{i>k} \mathit{store}(p_i),$$

with $\mathit{store}(p_i)$ as defined above and the empty conjunction set to true.

We leave it to the reader to check the correctness of this formula. Note that Proposition 9 states that a wff in the basic modal language can only force models with R–paths at most its modal depth. Now the modal depth of $C(n)$ is just two *for every* n, while the minimal R–depth of models satisfying $C(n)$ is $2^n - 1$. The difference is that $C(n)$ is a constraint which should hold globally.

2.6 Tree automata

The simplest way to see if a formula is satisfiable is to try to construct a model for it. This is what the K-World algorithm does. In fact it constructs a tree model if it can. Having constraints, constructing a tree model has to be done with care as it can become infinite (e.g., if the constraint is $\Diamond\top$). In effect one has to detect looping. Looping will occur as the tree is labeled with sets of relevant wffs (the Hintikka Sets) and there are just finitely many of them. We can see the Hintikka Set elimination algorithm as a particular way to implement this. Tree automata (in particular Büchi tree automata) are yet another way of dealing with infinite models. We describe those here and relate them to the other approaches. But before we go into them, we briefly look at a related problem: satisfiability on finite trees. Suppose we ask, given A and C, is A locally satisfiable in a *finite* tree that globally satisfies C? It seems not easy to adjust the Hintikka set elimination algorithm. We can use a tableaux approach (because we can compute a bound on the depth of the trees) but the complexity of it seems horrendous. As it turns out, a tree automaton approach has the right level of abstraction to deal with both infinite and finite trees in a unified manner. Chapter 17 describes the automata-theoretic approach to temporal reasoning.

A finite state automaton is a device to recognize (finite or infinite) strings in a given finite alphabet. Every automaton defines a language (a set of strings). The key result about these automata is a characterization theorem: a language L can be defined by a finite state automaton if and only if it can be defined by a regular expression. Recall the model $(\omega, \mathsf{succ}, V)$ of the logical system from the introduction of this section. Let V be restricted to Σ, a finite set of propositional variables. Then we can view this model as an infinite string in the alphabet $\mathcal{P}(\Sigma)$, indicating which variables are true at which worlds. The idea of model checking by automata is that for every formula A one creates an automaton A_A which accepts exactly those "models as strings" in which A is true at the origin.

But we can do more with A_A. We could try to check if A_A can accept any string at all. In other words, if A is locally satisfiable. This is called the emptiness problem of an automaton and for Büchi automata this problem is practically feasible.

Now finite state automata are fine for linear models, but in general models in modal logic are graphs. A number of interesting modal systems though have the tree model property. Tree automata generalize sequential automata in that they recognize ranked trees. So we can use the same idea as sketched above to decide satisfiability for the basic modal system using tree automata. In fact, tree automata might be used in any situation in which the models have a tree like property. Arguably the most powerful complexity results in modal logic –in particular for systems in which $\models$ has a second-order definition– have been obtained by employing tree automata.

We sketch the idea in a simple situation, satisfiability under constraints for the basic modal system. First come the technical notions. Let $[n]$ denote the set $\{1, \ldots, n\}$. An n-ary Σ-tree T is a labeling of the set $[n]^*$ by letters from an alphabet Σ. That is, $T : [n]^* \longrightarrow \Sigma$. The empty sequence ϵ is called the root of the tree.

A (non-deterministic) *Büchi automaton* on n-ary trees is a tuple $A = (\Sigma, S, \rho, S_o, F)$ where

- Σ is a finite alphabet.
- S is a finite set of states.
- $\rho : S \times \Sigma \longrightarrow \mathcal{P}(S^n)$ is the transition function. For each state $s \in S$ and letter $\sigma \in \Sigma$, it yields the set of possible S labellings of the n successors of state s.
- $S_0 \subseteq S$ is the set of initial states.
- $F \subseteq S$ is the set of final states.

A *run* of an automaton A on a tree T is a labeling of the nodes of T (notation $r : T \longrightarrow S$) by states S such that

- the root is labeled by an initial state (that is $r(\epsilon) \in S_0$), and
- the transitions obey the transition function ρ. That is, for each node x we have $\langle r(x \cdot 1), \ldots, r(x \cdot n)\rangle \in \rho(r(x), T(x))$.

A run $r : T \longrightarrow S$ is *accepting* if every branch of r visits F infinitely often. That is, for every branch $root, x_1, x_2, \ldots$ of T there are infinitely many i's such that $r(x_i) \in F$.

We will now show how Büchi automata relate to our earlier notions and how they are used to decide satisfiability. Let A, C be basic modal wffs. We have seen the following equivalences:

1. A is locally satisfiable in a model which globally satisfies C.
2. There exists a set $G \subseteq \mathcal{P}(Cl(A, C))$ satisfying conditions HS1–HS4.
3. A is satisfiable at the root of an n-ary tree which globally satisfies C. Here n is determined by the number of $\Diamond$ subformulas in A and C.

We can think of the tree in the last item as being labeled by elements of G (telling us immediately which subformulas of A and C are true in which worlds, without having to apply the truth definition). We call this tree model with the extra labels a *Hintikka tree* for A, C, if the root is labeled with a Hintikka set containing A. So we can add a fourth equivalent statement:

(iv) There exists a Hintikka tree for A, C.

Now our strategy must be clear. Given A and C we must build an automaton $A_{A,C}$ which takes n-ary trees labeled by subsets of $Cl(A, C)$ as input and accepts if and only if the input is a Hintikka tree for A, C. Then we have a fifth equivalent statement:

(v) The set of trees accepted by $A_{A,C}$ is not empty.

As checking emptiness can be done in polynomial time (in the size of the automaton), the complexity of the algorithm based on (v) depends on the complexity of the function creating $A_{A,C}$ from A and C. As we will see this takes time bounded by an exponential in $|A|$ and $|C|$. Thus the size of $A_{A,C}$ is similarly bounded and we obtain an exponential time decision algorithm.

We create $A_{A,C}$ as follows. Let G^- be the set of subsets of $Cl(A, C)$ satisfying HS1 and HS3. For simplicity we assume that C implies $\Diamond\top$, so that every model is infinite. This is easily lifted. Let the number of $\Diamond$ subformulas in A and C be n. $A_{A,C}$ is a Büchi automaton $(\Sigma, S, \rho, S_o, F)$ on n-ary trees with

- $\Sigma = S = F = G^-$.
- $S_0 = \{x \in G^- \mid A \in x\}$.
- ρ is defined as follows. $\rho(X, X)$ is the set of all sequences $\langle X_1, \ldots, X_n\rangle$ such that
  1. for all $\Diamond B \in X$, there is an X_i such that $B \in X_i$, and
  2. for all $\Diamond D \in Cl(A, C)$, for all X_i, if $D \in X_i$, then $\Diamond D \in X$.

  On all other inputs, ρ returns the empty set.

With all machinery developed so far it is straightforward to show that

$$A_{A,C} \text{ accepts a tree } T \text{ if and only if } T \text{ is a Hintikka tree for } A, C.$$

Thus the decision algorithm based on checking emptiness of $A_{A,C}$ is correct.

Now we show how we can check for finite tree satisfiability. Clearly we do not assume that C implies $\Diamond\top$ otherwise the problem is already solved. But we do add $\Diamond\top$ to the set $Cl(A, C)$. Then we create $A_{A,C}$ as follows:

- $\Sigma = S = G^-$.
- S_0 is as before.
- $F = \{x \in G^- \mid \Diamond\top \notin x\}$.
- ρ is defined for X containing $\Diamond\top$ as before and for $X \in F$ as $\rho(X, X) = \langle X, \ldots, X\rangle$. That is, the automaton gets into a self-loop once it reaches a state which should not have successors.

Now, this automaton accepts a tree if and only if T is a Hintikka tree for A, C and every node has a descendant which is the root of a subtree all of its nodes are labeled by one and the same set from F. In a real model we just cut off these subtrees leaving only their roots (which after all say that they do *not* have successors).

Vardi [39] has argued that the tree model property of modal logic is the reason for its robust decidability. Robust means here that expansions of the basic modal language with powerful possibly second-order operators does not destroy decidability. One way to show this is to show that the language is a fragment of the monadic second-order logic of trees, and use the decidability of the latter, which is proved using automata.

2.7 Pseudo-models

The models we constructed so far were not really pseudo-models, in fact they could all be considered as ordinary models. Now we look at two examples in which we shall be working with structures which look very much different from the intended models of the logics.

Two dimensional modal logic

The first logic that we consider here is —just as the tiling logic— also based on grid-like structures, but here we only require that the models are two-dimensional in nature, there will be no functions around. The language has two diamonds, $\Diamond_0$ and $\Diamond_1$, with the standard truth definition. The models are of the form $\mathcal{M} = (W, \equiv_0, \equiv_1, V)$, where we require that $(W, \equiv_0, \equiv_1)$ is in fact a *square* over some set U. That is, W consists of the set $U \times U$ of all *pairs* over U, while $s \equiv_i t$ holds if $s_i = t_i$: the i-th coordinate of s and the i-th coordinate of t should be the same. Denote the resulting system as $\mathbf{S5^2}$.

As a modal system, $\mathbf{S5^2}$ might look rather obscure, but as a logic it is well-known. In fact it is the exact modal counterpart of a restricted fragment of first-order logic with two variables in a signature having a binary relation symbol R for every propositional variable r. This is seen as follows. First observe that the $\mathbf{S5^2}$ model $\mathcal{M} = (W, \equiv_0, \equiv_1, V)$ with $W = U \times U$ is uniquely determined by the set U and the valuation V. Note that for any propositional variable r, $V(r) \subseteq U \times U$, i.e., a binary relation. Thus V can also be seen as an interpretation of the set of binary relation symbols R corresponding to each propositional variable, and we can view (U, V) as an ordinary first-order model. Also observe that we may identify assignments s mapping the two variables x_0 and x_1 to U with pairs $(s(x_1), s(x_0)) \in W$. Thus viewing the states of the modal models as assignments, we may read the statement 'φ holds in (U, V) under assignment s' modally as 'in model $(U \times U, \equiv_0, \equiv_1, V)$, φ is true at state s'. Because $\mathbf{S5^2}$ models are squares, the truth definition of the diamonds can be rewritten exactly as the definition of the first-order existential quantifiers:

$$\mathcal{M}, (a, b) \models \Diamond_0 \varphi \iff \text{there exists } a' \text{ such that } \mathcal{M}, (a', b) \models \varphi.$$

Thus $\Diamond_i$ is another way of writing $\exists x_i$. In a related way, one can define modal systems $\mathbf{S5^n}$ corresponding to first-order logic with n variables for any n. This is done in the field of cylindric modal logic, see [42, 29, 15].

Note that in this modal system, the local satisfiability and the satisfiability problem under constraints collapse. This is because A is locally satisfiable in a C model if and only iff $A \wedge \Box_0 \Box_1 C$ is locally satisfiable. Thus we consider the local satisfiability problem only. That problem is decidable, and a proof for this uses some kind of finite model property as well.

Here, instead of defining a finite model for A by *selecting* points out of the old model, we will *identify* points in the big model and define the finite model as a sort of quotient structure which we call –as before– *filtration* of the original model. It will turn out that this filtration will not be a square itself but a square-like structure that we here dub a *pseudo-square*. This is a model with an underlying frame (W, R_0, R_1) in which both R_0 and R_1 are equivalence relations, and their composition should be the universal relation. That is, (W, R_0, R_1) has to validate

(13) R_0 and R_1 are equivalence relations and $\forall xy \exists z (R_0 xz \wedge R_1 zy)$.

For these kind of structures, we can prove the following proposition. (The system also has the bounded finite model property with respect to squares, but this is much harder to establish

[31, 18].) As we saw before, decidability follows immediately, because it is decidable whether a finite structure is a pseudo-square (this is a first-order property).

PROPOSITION 11. *Any* $\mathbf{S5^2}$*-formula* A *is satisfiable in a square iff it is satisfiable in a pseudo-square of size not exceeding* $2^{|A|}$.

Proof. We concentrate on the left to right direction of this proof since we are only interested in explaining the notion of filtration to a pseudomodel at the moment. (For the other direction of the proof, one shows that given a pseudo-square model, one can always find a square that is bisimilar to it — in fact, bisimilar through a *functional* bisimulation, see [29]).

Suppose that A is satisfied somewhere in the square model $\mathcal{M} = (W, \equiv_0, \equiv_1, V)$. From this we will prove that A is true somewhere in a *filtration* $\mathcal{M}^f$ of $\mathcal{M}$. As before, as the domain of $\mathcal{M}^f$ we take the set of theories:

$$W^f = \{\theta_{\mathcal{M}}^{Cl(A)}(w) \mid w \text{ in } \mathcal{M}\}.$$

Again we want to prove a truth lemma, so the valuation is now also fixed. What would be a good definition for the relations R_0 and R_1 on W^f? In general, this is where the filtration method needs some creative input. Now, if the only requirement were that A is to be true somewhere in the resulting model, there is a whole family of definitions that work (in the sense that they ensure that the conditions **min** and **max** are satisfied). But the extra constraint, viz., that the resulting model should be a pseudo-square, puts some extra restrictions. In any case, the following definition works:

$$R_i(\theta_{\mathcal{M}}^{Cl(A)}(s), \theta_{\mathcal{M}}^{Cl(A)}(t)) \text{ if for all } \Diamond_i\varphi \in Cl(A)\text{: } \mathcal{M}, s \models \Diamond_i\varphi \text{ iff } \mathcal{M}, t \models \Diamond_i\varphi.$$

(The reader should check that this is well-defined.)
We can now prove the main claim concerning filtrations, the truth lemma:

(14) for all formulas $\varphi \in Cl(A)$, for all $\theta_{\mathcal{M}}^{Cl(A)}(s)$: $\mathcal{M}^f, \theta_{\mathcal{M}}^{Cl(A)}(s) \models \varphi$ iff $\varphi \in \theta_{\mathcal{M}}^{Cl(A)}(s)$.

This claim is proved by a formula induction as before. For the diamond cases, one should check that **min** and **max** indeed hold for the given definition of the relations. For **min**, use the fact that $\varphi \to \Diamond_i\varphi$ is valid for every φ in this logic (because $\equiv_i$ is a reflexive relation). For **max**, use the fact that $s \equiv_i t$ implies that $\mathcal{M}, s \models \Diamond_i\varphi$ iff $\mathcal{M}, t \models \Diamond_i\varphi$.

This proves (14), so in order to prove the left to right direction of the Proposition we only have to show that $\mathcal{M}^f$ is a pseudo-square. We leave it to the reader to verify that both R_0 and R_1 are equivalence relations. In order to check the other condition, consider sets $\theta_{\mathcal{M}}^{Cl(A)}(s)$ and $\theta_{\mathcal{M}}^{Cl(A)}(t)$. Now the fact that $\mathcal{M}$ is a square and that s and t are pairs comes in handy. Let $z = (s_0, t_1)$. Then $s \equiv_0 z \equiv_1 t$. But then it follows that $R_0(\theta_{\mathcal{M}}^{Cl(A)}(s), \theta_{\mathcal{M}}^{Cl(A)}(z))$ and $R_1(\theta_{\mathcal{M}}^{Cl(A)}(z), \theta_{\mathcal{M}}^{Cl(A)}(t))$, which shows that indeed, the composition of R_0 and R_1 is the universal relation on $\mathcal{M}^f$. ❑

The reader might wonder whether we can construct the pseudosquare by a Hintikka Set elimination procedure as well, just as we did with the basic modal system. The only difference is that we have to end up in a pseudosquare, whereas with the basic modal system any model was allowed. But how to implement the check for the condition $\forall xy \exists z(R_0xz \wedge R_1zy)$? Suppose it fails for some Hintikka sets, x, y. Which one should we remove from the set of candidates? It seems we have to consider both possibilities. But then we get an algorithm which takes far longer than the original Hintikka set elimination procedure. Instead of $2^{|A|}$ many stages we have $2^{2^{|A|}}$ many. In section 3.5 we will show that indeed this is a harder problem than the satisfiability problem under constraints for the basic modal system.

The until operator

We now consider the modal system given by the propositional language expanded with the binary until operator U, the class of all models of the form (W, R, V) and an interpretation of U as in the chapter on temporal logic (recalled below). Note that this is not a modal system in the strict sense of Chapter 5 of this handbook, as U has a dual existential–universal definition: $\mathcal{M}, s \models \mathsf{U}(\varphi, \psi)$ if and only if there exists u such that $R(s, u)$ and $\mathcal{M}, u \models \varphi$ and for all t such that $R(s, t)$ and $R(t, u)$ it holds that $\mathcal{M}, t \models \psi$.

Our previous methods do not work for this language as they were based on reasoning on trees. It is not hard to see that the formula $\mathsf{U}(p, \top) \wedge \neg\mathsf{U}(p, p)$ is satisfiable but not in a tree. Nevertheless, we will show that this system does have a finite pseudo-model property, and we use this property for showing that it has a decidable satisfiability problem.

To start with, let $\mathcal{L}_{\mathsf{U}}$ be the language obtained by expanding the classical propositional language with the binary connective U. It is convenient to use the following notation: for s and u elements of the domain of some model W,

(15) $\mathcal{M}, su \models \psi$ iff for all t satisfying Rst and $Rtu, \mathcal{M}, t \models \psi$,

because we can now rephrase the truth definition of the until operator as follows:

(16) $\mathcal{M}, s \models \mathsf{U}(\varphi, \psi)$ iff for some u such that Rsu $\mathcal{M}, u \models \varphi$ and $\mathcal{M}, su \models \psi$.

Let M denote the class of all models (W, R, V). We call the resulting modal system $(\mathcal{L}_{\mathsf{U}}, \mathsf{M}, \models)$ the *until system*. Note that $\mathcal{L}_{\mathsf{U}}$ is normally interpreted on models in which R is a linear order, but here we disregard this extra complicating factor.

Different from our earlier proofs, we will not use any kind of finite *model* property in order to prove decidability for the until system. This is not because the system does not have the bounded finite model property (it does); our proof method is for didactic purposes. The idea behind the *mosaic method* is that we construct a finite pseudo-model that we will call a linked set of mosaics. One then has to show that a formula is satisfiable if and only if there exists such a linked set of mosaics for it.

What then are mosaics? One could best describe them as little pieces of a model that, if linked together in a nice way, contain sufficient information to *construct* a real model.

Concerning the notion of a mosaic then, the first question is *what information* we are interested in. This question is easy to answer: as in all previous proofs we are only interested in the truth of subformulas of A. The second question then should be: how large should the little pieces of model be? In all previous proofs, the parts we worked with consisted of just one world. The definition of the until operator makes that this is not enough. We need to define a new concept. Call a subset of the domain of a model $\mathcal{M} = (W, R, V)$ *packed* if every two distinct elements s and t of the subset are R-related (that is, we require that Rst or Rts). Our patchwork pieces then will be packed sets of size at most three.

The number three here derives from the fact that the truth definition of $\mathsf{U}(\varphi, \psi)$ employs three variables. In fact, if one would try to devise a standard translation or a bisimulation game for the $\mathcal{L}_{\mathsf{U}}$-language, the number three would show up as the minimal number of variables needed and as the minimal size of the windows that cover the models during the game. During a game one would see that these windows will only be placed on packed sets of the models. Note that in a tree model there are *no* packed sets of size three, only of size two. Also note that in the two dimensional logic the packed sets in a model are exactly the rows and the columns.

Abstracting from the origin of these pieces, we arrive at the following definition. From now on we let A be an arbitrary but fixed $\mathcal{L}_{\mathsf{U}}$ formula. A is the formula whose satisfiability needs

to be decided. As usual, we let $Cl(A)$ denote the closure under single negations of the set of subformulas of A.

DEFINITION 12. An A-type *mosaic* is a quadruple $\mu = (X, R, A_\varphi, B_\varphi)_{\varphi \in Cl(A)}$ such that X is a set of size at most three; R and every B_φ are binary relations on X; and every A_φ is a unary relation on X. When A is clear from the context, we just use "mosaic".

The basic idea underlying this definition is that A_φ holds of a point if we 'want' φ to be true at it, while B_φ holds of a pair of points if we 'want' φ to be true at each point between them. Obviously, not every such structure is part of some model — we need some further constraints for that. Call a mosaic *coherent* if it satisfies the following conditions (phrased in first-order logic and to be read universally):

$(C0)$ $Rxy \vee Ryx \vee x = y$
$(C1)$ $A_{\neg\varphi}x \leftrightarrow \neg A_\varphi x$
$(C2)$ $A_{\varphi\wedge\psi}x \leftrightarrow A_\varphi x \wedge A_\psi x$
$(C3)$ $B_{\varphi\wedge\psi}xy \leftrightarrow B_\varphi xy \wedge B_\psi xy$
$(C4)$ $(Rxy \wedge Ryz \wedge B_\varphi xz) \rightarrow A_\varphi y$
$(C5)$ $(Rxy \wedge A_\varphi y \wedge B_\psi xy) \rightarrow A_{\mathsf{U}(\varphi,\psi)}x$

A few words of explanation: C0 reflects the fact that we only took packed subsets of the model as the domain of our mosaic mini-models. C1–C3 are self-explanatory; note that there is no analog of C1 for the B-predicates since there is a hidden universal quantifier in the meaning of a predicate B_φ, cf. (15). Finally, C4 and C5 are rather obvious consequences of our intuitive meaning of the A- and B-predicates and the truth definition of the until operator.

How difficult is it to check whether an A–type mosaic is coherent, measured as a function of $|A|$? The size of an A–type mosaic is bounded by a polynomial in $|A|$. The length of the first-order formula formalizing coherence is bounded by a polynomial in $|A|$, and is written using just three (again!) variables. So this check can be made using $p(|A|)$ steps for some polynomial p.

The conditions C0–C5 take care of all *universal* constraints on the A- and B-predicates; but of course there are *existential* demands as well which we will call *requirements*. A requirement of a mosaic $\mu = (X, R, A_\varphi, B_\varphi)_{\varphi \in Cl(A)}$ is one of the two following types of object: for $s, t \in X$

(a) $(A_{\mathsf{U}(\varphi,\psi)}, s)$ such that $A_{\mathsf{U}(\varphi,\psi)}s$,
(b) $(notB_\varphi, s, t)$ such that Rst and *not* $B_\varphi st$.

In order to explain requirements of type (a), suppose that we want the formula $\mathsf{U}(\varphi, \psi)$ to be true at a point s; if there is a point t in the mosaic such that Rst, $A_\varphi t$ and $B_\psi st$, then the mosaic itself *directly fulfills* the requirement (by $(C5)$). This will rarely be the case however; the whole point of the mosaic method is that requirements can be fulfilled by *distinct* mosaics as well, as follows. A *link* between two mosaics μ and μ' is simply a partial isomorphism between the two structures. We say that a link $f : \mu \hookrightarrow \mu'$ *fulfills the requirement* $(A_{\mathsf{U}(\varphi,\psi)}, s)$ of μ if there is some t in μ' with $Rf(s)t$, $A_\varphi t$ and $B_\psi f(s)t$. Likewise, a link $f : \mu \hookrightarrow \mu'$ *fulfills the requirement* $(notB_\varphi, s, t)$ if there is some u in μ' with $Rf(s)u$, $Ruf(t)$ and $\neg A_\varphi u$.

A collection L of mosaics is called a *linked set of mosaics* if every requirement of every mosaic $\mu \in L$ is fulfilled via some link $f : \mu \hookrightarrow \mu'$ to some μ' also in L. It is a linked set of mosaics *for* A if it contains a mosaic with non-empty A_A.

Given a collection L of A-type mosaics, how difficult is it to check that it is a linked set of mosaics *for* A? We can view such a collection L as one first-order structure in the same signature as A-type mosaics in which all the mosaics are pairwise disjoint. Then the universe of L is bounded by three times the number of mosaics. Checking coherence of all mosaics can be done using $|L|$ checks each taking $p(|A|)$ computation steps, for a polynomial p. That L is a

linked set of mosaics for A is a simple first-order statement: $\exists x A_A(x)$. But also the existence of partial isomorphisms can be expressed in first-order logic, by a formula in five variables whose length is polynomially bounded by $|A|$. So the check can be made using a polynomial number of steps in $|L|$ and $|A|$.

Now the main result concerning mosaics is the following.

PROPOSITION 13. *An $\mathcal{L}_{\mathsf{U}}$-formula A is satisfiable if and only if there is a linked set of mosaics for A.*

THEOREM 14. *It is decidable in $2^{O(|A|)}$ steps whether an $\mathcal{L}_{\mathsf{U}}$-formula A is satisfiable.*

Proof. We can adjust the Hintikka set elimination algorithm given for the system **K** in order to deal with mosaics. This is done as follows. Let S be the set of all A-type mosaics (up to isomorphism). It is not hard to show that $|S| \leq 2^{O(|A|)}$. Let $S_0 \subseteq S$ be the subset containing all coherent mosaics. Thus S_0 can be computed in polynomial time in $|S| \leq 2^{O(|A|)}$ and $|A|$. We now inductively construct a sequence of sets of mosaics $S_0 \supsetneq S_1 \supsetneq S_2 \supsetneq S_3 \cdots$, just as in the proof for the system $\mathbf{K}^*$. The idea is that we delete mosaics from S_i if they have a requirement which can not be fulfilled inside S_i. We already showed that the checks to be made at each stage of the algorithm take time polynomial in the size of S_i and $|A|$. As $|S| \leq 2^{O(|A|)}$, there are at most $2^{O(|A|)}$ stages, thus the whole algorithm can be performed in $2^{O(|A|)}$ steps. ❑

We end with the rather involved proof of the correctness of the algorithm.

Proof of Proposition 13 The left to right direction of the proof is easy: suppose that $\mathcal{M} = (W, R, V)$ is a model for A. Out of this model we will cut a linked set of mosaics for A, as follows. Let P be the collection of all packed subsets of W of size at most three. Associate with any set $X \in P$ a mosaic μ_X based on the set X, with R as in $\mathcal{M}$ and with every A_φ and B_φ defined as given by the truth of φ in $\mathcal{M}$. We leave it as an exercise for the reader to verify that the collection of all these mosaics forms indeed a linked set of mosaics.

The direction from right to left in the Proposition is the hard one, although the key idea underlying its proof is quite intuitive. We will construct a model for A *step by step*; that is, we will approximate our model via a series of finite structures that we call *networks*. A *network* is a structure $\mathcal{N} = (W, R, A_\varphi, B_\varphi)_{\varphi \in Cl(A)}$ of the same type as a mosaic but not bounded in size. A network is called *coherent* if it satisfies the conditions C1–C5 above. To ask for C0 would be too much; instead we require coherent networks $\mathcal{N}$ to satisfy the following:

(liveness) every packed set X of size at most three *comes from* a mosaic; that is, for each such set $X \subseteq W$ there is a partial isomorphism $f : \mathcal{N} \hookrightarrow \mu$ such that f is defined on X.

Liveness means that — through the mosaics — we are in control of certain small parts of the model: the packed sets of size at most three. Why only these sets? The truth definition of U provides the answer. The meaning of $\mathsf{U}(\varphi, \psi)$ depends only on these small packed sets in the model.

A *defect* of a network is a requirement that is not directly fulfilled in the network itself, and a network is called *saturated* if it has no defects. A network is *perfect* if it is both coherent and saturated.

This name is well-chosen, since perfect networks are the ones that we are after. The reason for this is that with every network $\mathcal{N} = (W, R, A_\varphi, B_\varphi)_{\varphi \in Cl(A)}$ we can associate a modal model in an obvious way: it is defined as the structure $\mathcal{N}^\circ = (W, R, V^\circ)$ with $V^\circ(p) = A_p$ for all variables p occurring in A. But only for perfect networks can we prove the following *truth lemma*.

CLAIM 15. If $\mathcal{N}$ is a perfect network, then for all formulas $\varphi \in Cl(A)$ and all points s, t in $\mathcal{N}$:

1. $s \in A_\varphi$ iff $\mathcal{N}^\circ, s \models \varphi$,
2. if Rst, then $(s,t) \in B_\varphi$ iff $\mathcal{N}^\circ, st \models \varphi$.

PROOF OF CLAIM The proof of this claim is by induction on the complexity of φ. We only consider the case where φ is of the form $\mathsf{U}(\psi, \chi)$, and only prove part (ii) of the Claim (the first part is simpler).

By the induction hypothesis and the truth definition of U, in order to prove (ii) it suffices to show that for all pairs of points s and t such that Rst, we have that $(s,t) \notin B_\varphi$ iff $u \notin A_\varphi$ for some u with Rsu and Rut. The left to right direction immediately follows from the fact that $\mathcal{N}$ is perfect and thus all requirements of type (b) are fulfilled. For the other direction, suppose that s, t and u are points satisfying Rst, Rsu, Rtu and $u \notin A_\varphi$. Observe that $\{s, t, u\}$ is a packed set of size at most three, whence we may use the (liveness) condition. This yields a partial isomorphism f from $\mathcal{N}$ to some mosaic μ such that f is defined for each of s, t and u. It follows that $Rf(s)f(t)$, $Rf(s)f(u)$, $Rf(t)f(u)$ and $f(u) \notin A_\varphi$; but then it follows from condition C4 that $(f(s), f(t)) \notin B_\varphi$. Returning to $\mathcal{N}$ this shows that $(s,t) \notin B_\varphi$, which is what we needed to prove. ❑

From the previous claim it follows that in order to show that A is satisfiable, it suffices to show that there is a perfect network for it, that is, a perfect network such that A_A is not empty.

CLAIM 16. There is a perfect network for A.

PROOF OF CLAIM The proof of this claim consists of three parts. First we show that there is *some* network for A (not necessarily perfect). This is easy, since we are given a linked set of mosaics for A: as our network we simply take any mosaic with non-empty A_A.

The second and main part of the proof consists in showing that any defect of any network can be *repaired*; that is, we can find a bigger network in which the defect no longer occurs. Without going too much into technical detail, let us see how to repair a defect of type (b) (defects of type (a) are repaired in a similar way).

Suppose that s and t are points of the network $\mathcal{N}$ such that Rst and *not* $B_\varphi st$ for some subformula φ of A, while there is no point u between s and t such that $\neg A_\varphi u$. The idea now is simply to *repair* this defect by adding a *new point* to the network. What kind of point? Well, since we have Rst we know that s and t come from a mosaic; that is, there is a partial isomorphism f from $\mathcal{N}$ to some mosaic μ. Obviously, $(notB_\varphi, f(s), f(t))$ is a requirement of this mosaic. But since we are working in a *linked* set of mosaics, there must be some link g between μ and μ' and some u in μ' such that $Rg(f(s))u$, $Rug(f(t))$ and $\neg A_\varphi u$. Now simply add an entirely new point r to the network; make sure that the relations between s, t and r are such that this part of the model is isomorphic to μ'. It is thus obvious that we have *repaired* the defect, and that the new structure is a network. In order to keep the liveness condition it is essential *not* to relate r to any *other* point besides s and t: in this way the only new packed sets are $\{r, s, t\}$ and its subsets.

Finally, these two parts provide the material and the tools for constructing the desired perfect network for A. Starting from the mosaic for A (which is of course a network), we repair defects, one by one, step by step, thus constructing a sequence $\mathcal{N}_0$, $\mathcal{N}_1$, ... of networks. Using some standard combinatorics we can ensure that the *limit* of the chain of networks is a network without defects. In particular, if we always take new points from a fixed set, say ω, we can enumerate the set of all (potential) defects of any network of the chain; if at each step of the construction we

repair the current network's defect with the lowest number in this enumeration, we can create a perfect network. ❑

3 COMPLEXITY

[27] lists the following as the basic questions of computer science: What is an algorithm? What can and what cannot be computed? When should an algorithm be considered practically feasible? In the previous sections we have —sometimes between the lines— used the available answers to these questions. Before we go back to modal logic we catch a brief glimpse of the theories of computability and computational complexity. Readers familar with this can jump immediately to subsection 3.5.

3.1 Computability

What is an algorithm? It is much easier to decide that a certain procedure can be labeled an algorithm than to give a definition. But a definition is needed when we want to prove that *no* algorithm exists to decide a certain problem. Recall that is what we did with the tiling logic in Section 1.3. Such a definition should be very robust. We would not like that later someone comes up with a procedure for deciding satisfiability under constraints for the tiling logic which we have to accept as an algorithm. As it turned out, the first proposal, now called Turing machine, named after its inventor Alan Turing, was immediately right. For a definition of Turing machines see any complexity or finite model theory textbook or Wikipedia. Every other model of computation that has been defined up to now has been shown to be equivalent to the Turing machine model.

The so-called Church Turing thesis, turns these empirical facts into a principle:

Church–Turing Thesis. The Turing machine that terminates on all inputs is the precise formal notion corresponding to the intuitive notion of an algorithm.

More precisely, for Σ a finite alphabet and $S \subset \Sigma^*$ a set of Σ-strings, there exists an algorithm for deciding membership of Σ-strings in S if and only if there exists a Turing machine which terminates on all input strings $s \in \Sigma^*$ and correctly outputs the answer to the question whether $s \in S$.

The reader may substitute his preferred sufficiently powerful programming language for Turing machine in the thesis (as each sufficiently powerful language is equivalent in computation power to the Turing machine model). Note that the thesis speaks about *all inputs*. That is the reason why we had to be so careful in Section 1 when we spoke about the desirable properties of a logic.

Everyone with a limited amount of programming experience has created programs which do not terminate on some inputs. Such programs are not useful for deciding a yes/no problem, simply because for the particular input on which it does not terminate we do not get an answer. This is a good place to discuss the notion of *semi-decision*. A yes/no problem is said to be semi-decidable if a program (a Turing machine) exists which terminates on all yes instances and on all no instances it does not terminate. The set of valid first-order sentences is the prime example of such a set. It is semi-decidable but not decidable. Note that if both the yes and the no instances (that is, both "halves" of the problem) are semi-decidable, we can decide the whole language.

The reason is that on any input we can start *both* programs, one which terminates on the yes, and the other which terminates on the no-instances. Whatever instance we have, one program must terminate, so we have a decision procedure. Thus a problem is decidable iff both halves of it are semi-decidable. See [27] for a detailed argument.

This is all we want to say about computability. The Church-Turing thesis yields an exact measure between the computable and the non computable problems. There is theory about the difficulty of a problem regardless whether it is a computable or a non computable problem. For instance, we just saw that first-order validity is not computable but still semi-decidable. A natural question is whether there exist even harder problems: those that cannot even be semi-decided. Indeed they do exist, and tiling problems have been developed for these cases as well. For instance, add to the tiling problem an extra special tile T^* and ask whether a tiling exists such that T^* occurs infinitely often on the first row. The complement of our original tiling problem is semi-decidable (Why? Because we can formalize it in terms of first-order logic!), but this is not true anymore for this extended version. Cf., [22] for further reading.

As this chapter is about complexity we now take a closer look at the computable problems.

3.2 Computational complexity

In Section 1 we spoke about practically feasible algorithms, and defined them to be those which take at most $p(n)$ computation steps, for p a polynomial function in n with n representing the length of the input. We were rather vague on what constitutes a computation step, but we can now make that precise using the Turing machine model: it is a step taken by a Turing machine. Even inside the practically feasible and inside the not practically feasible algorithms it makes sense to try to distinguish problems of different complexity. As we are mostly concerned with non practically feasible problems we take a closer look there. Let us first give a formal enough definition of the class **P** of problems solvable in polynomial time.

DEFINITION 17. A Turing machine is polynomially time bounded if there is a polynomial $p(n)$ such that the machine always halts after at most $p(n)$ steps, where n is the length of the input. A problem is solvable in polynomial time (a function is computable in polynomial time) if there is a polynomially time bounded Turing machine that solves it (that computes it). The class of all problems solvable in polynomial time is called **P**.

Within the class of non practically feasible algorithms we have those which take exponential time. We define the class **EXPTIME** of problems solvable in exponential time by requiring that there is some polynomial p such that the Turing machine must halt on all inputs after at most $2^{p(n)}$ steps. Similarly we can define the class of **2EXPTIME** problems, solvable in at most $2^{2^{p(n)}}$ steps, and so on. These classes form a hierarchy and it is known that all inclusions are strict. Consider the class **EXPTIME**. Can we find interesting and natural subclasses here? We can look at the amount of working memory a Turing machine needs in its computation. At our level of discussion it is enough to know that in every computation step of a Turing machine, the machine can read and write a symbol of a specified finite alphabet in a cell of a tape. The tape corresponds to the memory of the Turing machine. Clearly machines are conceivable which take a long time before they reach their decision but which use only a limited amount of memory. A good example is the "bad way of doing model checking" described in Chapter 1. We can define space classes similarly to time classes:

DEFINITION 18. A Turing machine is polynomially space bounded if there is a polynomial $p(n)$ such that no computation of the machine scans more than $p(n)$ tape cells. The class of all

problems solvable by a polynomially space bounded Turing machine is called **PSPACE**.

It should be clear that **P** is contained in **PSPACE**. Note that there is no bound on the amount of time for polynomially space bounded computations. Still we have

PSPACE is contained in **EXPTIME**.

To see this, consider the number of ways a tape containing $p(n)$ cells can be written, in an alphabet containing two symbols. That is $2^{p(n)}$. This bounds the number of configurations of a Turing machine. If a **PSPACE** machine takes more than exponential time, then it has repeated some configuration so it must be in an infinite loop. This is not possible because it must terminate on each input.

Non-determinism. The Turing machine model we had in mind was the *deterministic* Turing machine. This is what we typically think of when we think of a computer program: the output state of a computer program is uniquely determined by its input state (that is, by its input). Moreover, for *every intermediate* state there is exactly one state in which the program can be at the next computation step. In essence, a program is a function from input states to output states.

One of the more difficult abstractions made of a Turing machine is a *non-deterministic* Turing machine. Non-deterministic programs are very useful when the problem consists of a (difficult) search for a solution and an (easy) check whether the solution is correct. The prime example of such a problem is boolean satisfiability: given a propositional formula, does there exists a valuation of the proposition letters such that the formula evaluates to true? For a formula with n letters there are 2^n many possible valuations, an enormous search space. To check whether a given valuation evaluates to true is very easy (the number of steps is polynomially bounded by the number of connectives in the formula). We can write a non-deterministic program which scans the formula from left to right, and has the following non-deterministic rule

> if you read p_i for the first time, then either replace all occurrances of p_i by `true` or replace all occurrances of p_i by `false`.

Consider the application of this rule as one computation step. After scanning the complete formula the program can simply check whether the result evaluates to true. Thus if a formula is satisfiable the program can answer fast and correctly. If the formula is not satisfiable, then none of the choices will lead to a state in which the program will answer that it is. So this program decides the satisfiability problem.

We say that a non-deterministic Turing machine decides a problem if

- for every yes instance of the problem, there is *at least one* computation that accepts the input, and
- for every no instance, *every* computation rejects the input.

Note the asymmetry in this definition. Deterministic Turing machines have just one computation on each given input: this is exactly what determinism requires. But the above non-deterministic program has 2^n many different computations on a formula with n variables. Each of them leads to an accepting or rejecting state in $p(n)$ steps, for p a polynomial. This is the time we measure.

DEFINITION 19. **NP** is the class of problems decided by a polynomially time bounded non-deterministic Turing machine. **NEXPTIME** is the class of problems decided by an exponentially time bounded non-deterministic Turing machine.

Still the reader might ask how long a non-deterministic polynomially time bounded computation really takes. Where "really" means implemented on a deterministic Turing machine. This is one of the greatest puzzles in computer science. The answer is simply:

> *At present we can only do it using an exponentially time bounded deterministic Turing machine. We do not know if we can do better, but we consider it unlikely.*

In other words, the question whether **P**≠**NP** is wide open, though it is generally believed that the two classes are different. Thus we have the following inclusions

$$\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE} \subseteq \mathbf{EXPTIME} \subseteq \mathbf{NEXPTIME},$$

and all we know at present is that $(\mathbf{N})\mathbf{P} \subsetneq (\mathbf{N})\mathbf{EXPTIME}$.

3.3 The complexity of modal decision problems

In Section 2 we gave several decision algorithms and analyzed their complexity in rather loose terms. In fact these arguments can be made rigorous without too much ado. Thus we can use them to argue that the problems discussed earlier are in certain complexity classes. We briefly list the results:

1. Propositional satisfiability is in **NP**.
2. Local satisfiability of the logical system tile is in **NP**.
3. Local satisfiability of the basic modal system **K** is in **PSPACE**.
4. Satisfiability under constraints of the basic modal system **K** is in **EXPTIME**.
5. Local satisfiability of two dimensional modal logic is in **NEXPTIME**.

These are nice results but they do not tell us very much, except that for none of them we have found a practically feasible algorithm. Surely that does not mean that no such algorithm exists. We would like to have a similar situation as with decidability: either we show a problem is decidable or we show it is undecidable. Thus we would like to be able to show that a problem is *not in* ***P*** , or it is in **PSPACE**, but not in **NP**, etc. Unfortunately, except for some problems in **EXPTIME**, we cannot show that a problem is not in **P**. At present we just do not know how to do that. As a second best to showing that a problem in **NP** is not in **P** we can show that it is a *core problem* in **NP**. A really desirable property of a core problem Q in **NP** would be that

if Q happens to be in **P**, then every problem in **NP** is also in **P**.

In other words, if you are able to find a polynomial time algorithm for Q, you have found one for all problems in **NP**. Even though this property sounds extremely strong, we now have hundreds of core problems in **NP** (cf., [16]). These are called **NP** complete problems and the crucial notion in their definition is that of a polynomial time reduction.

3.4 Reductions

We have used a reduction from an undecidable problem (tiling) to another problem to show that the latter is also undecidable. The reduction used in the proof of Theorem 7 took an instance of the tiling problem and produced two formulas A and C. The crucial point of the reduction was that it was a *computable* function. Computability was all we needed to transfer undecidability.

In the case at hand we want to transfer a similar negative property: most likely not in **P**. But then we should not ask for mere computability, but for computability in polynomial time. Note that the given reduction of the tiling problem is polynomial time computable. It is an example of a *polynomial reduction* of one problem into another. This simple and intuitive notion is very powerful. To recapitulate: if there is a polynomial reduction from problem A to problem B, then B is at least as hard as A. If we can solve B efficiently (that is, if $B \in \mathbf{P}$), then the same holds for A. But if A requires exponential time, then so does B.

DEFINITION 20. A problem Q is said to be **NP** complete if Q is in **NP** and for every problem Q' in **NP** there is a polynomial reduction from Q' to Q.

We invite the reader to show that **NP** complete problems have the desired property of core problems discussed above. We define complete problems for the other complexity classes in exactly the same way, all the time using polynomial reductions.

From the definition it looks very difficult to show that a given problem which is in **NP** is also **NP** complete, because of the universal quantifier. But that is just appearance, as any problem in **NP** can be decided by a non deterministic polynomially time bounded Turing machine. Thus we need to reduce the acceptance problem of such machines. The first problem shown **NP** complete was boolean satisfiability (This is known as Cook's Theorem.) Once you have one **NP** complete problem it is much easier to get more. Because the composition of two polynomial reductions is a polynomial reduction all one needs to do to establish **NP** completeness of a problem Q in **NP** is to reduce a known **NP** complete problem to it. Similarly for the other complexity classes.

Thus when doing a complexity analysis of a problem one should try to establish a matching upper and lower complexity bound: that is, establish that the problem is complete for some complexity class. The upper bound indicates the amount of time and space needed by the best known algorithm. The matching lower bound that —with the present state of knowledge in computer science— one cannot do better. Note that "cannot do better" is a rather relative notion. For one thing, both $10 \cdot n$ and n^{10} are polynomial functions. As none of the problems we look at is solvable by a practically feasible algorithm anyway, it is at this stage of the discussion enough to use the rather broad but robust complexity classes we discussed.

We will now establish those required lower bounds for the problems listed in the beginning of Section 3.3. When we have reduced a known **NP** complete problem to a problem Q we say that Q is **NP** hard, and so on. Such reductions yield good insight in the expressive power and the design of the modal system.

3.5 Tiling

We will establish matching lower bounds by reductions to finite versions of the tiling problem discussed in Section 1.3.

We start with the simplest finite version, square tiling. The problem is as before, we are given a set of tiles T, containing a special tile T_0, and a natural number n. Note that n is part of the input. The problem is whether the $n \times n$ square can be tiled using the tiles T with the constraints

that T_0 is placed at the four borders on the square. Interestingly, the complexity of this problem depends on how the size n is given.

FACT 21.

1. The $n \times n$ tiling problem with n given in unary is NP complete.
2. When n is given in binary it is NEXPTIME complete.

To get in the mood let us show that propositional satisfiability is NP hard, by reducing the tiling problem to it. So consider an instance of the problem consisting of an n and a set of tiles $T = \{T_0, T_1 \ldots, T_k\}$. We want to create a propositional formula A^n_T which is satisfiable iff there exists a T tiling of the $n \times n$ square. Moreover the formula A^n_T must be computable in polynomial time from T and n. A^n_T will be constructed from the set of propositional variables

$$\{p^t_{xy} \mid 0 \leq t \leq k,\ 0 \leq x, y < n\}.$$

The intended meaning of p^t_{xy} is "Tile T_t is placed on position (x, y)". Note that there are $n^2 \cdot (k+1)$ variables p^t_{xy}. A^n_T is a conjunction of clauses. It consists of the following parts:

- At each point in the square, exactly one tile is placed:

$$\begin{array}{ll} \text{(at least one)} & \bigwedge_{0 \leq x,y \leq n-1} (p^0_{xy} \vee \ldots \vee p^k_{xy}) \\ \text{(at most one)} & \bigwedge_{\substack{0 \leq x,y \leq n-1 \\ 0 \leq t \neq t' \leq k}} \neg(p^t_{xy} \wedge p^{t'}_{xy}). \end{array}$$

- The special tile T_0 is placed all along the edges of the square:

$$\text{(border)} \qquad \bigwedge_{0 \leq i \leq n-1} (p^0_{0i} \wedge p^0_{n-1i} \wedge p^0_{i0} \wedge p^0_{in-1}).$$

- Colors must match horizontally and vertically:

$$\begin{array}{ll} \text{(horizontal)} & \bigwedge_{\substack{0 \leq x,y < n-1 \\ 0 \leq t \leq k}} (p^t_{xy} \rightarrow \bigvee_{t' \mid \mathrm{right}(t) = \mathrm{left}(t')} p^{t'}_{x+1y}) \\ \text{(vertical)} & \bigwedge_{\substack{0 \leq x,y < n-1 \\ 0 \leq t \leq k}} (p^t_{xy} \rightarrow \bigvee_{t' \mid \mathrm{up}(t) = \mathrm{down}(t')} p^{t'}_{xy+1}). \end{array}$$

Clearly this formula is computable from the instance. It is also not hard to show that it is a correct reduction: the formula is satisfiable if and only if the instance can tile the given square. But in order to show that we have given a polynomial reduction, we need to argue that the formula is polynomially computable from the given tiling instance. What is the size of the formula? Look at the last conjunct (vertical). It consists of $n^2 \cdot (k+1)$ conjunctions each containing at most $k+2$ variables and $k+1$ connectives. What is the size of a propositional variable? Naively we think of propositional variables as letters $p, q, r \ldots$ so it is tempting to assume that they have size 1. But if we encode variables in some standard way, e.g. as strings of zeroes and ones, then the more we have of them the more bits we need. We always assume that we use a binary encoding, so for our $n^2 \cdot (k+1)$ many variables we need $\log(n)^2 \cdot \log(k+1)$ bits for each variable. Thus the total size of (vertical) is $O(n^2 \cdot \log(n)^2 \cdot k^2 \cdot \log(k))$. We obtain similar bounds for the other conjuncts. Note how lucky we are that n was given in unary. Otherwise, when n had been given in binary, we had already $2^n \cdot 2^n \cdot (k+1)$ many propositional variables, so our reduction would never have been polynomial.

We will use the binary version of square tiling later on. First we look at a variant called corridor tiling.

Corridor tiling. Given a set of tiles T, a special tile T_0 and an integer n in unary. The problem is whether there exists a height m such that the rectangle $n \times m$ can be tiled by T, with all borders tiled by T_0.

FACT 22. Corridor tiling is PSPACE complete.

Corridor tiling is very well-suited to obtain results for linear structures. Consider the logical system succ consisting of the basic modal language, its models are all initial segments of the natural numbers with successor, and $\Diamond$ is interpreted by the successor function.

PROPOSITION 23. *The problem whether A is (locally) satisfiable in a model which globally satisfies C is PSPACE hard for the system* succ.

Proof. Of course we reduce corridor tiling. Let $T = \{T_0, \ldots, T_k\}$ and n determine an instance. We use variables p_x^t for $0 \leq x \leq n-1$ and $0 \leq t \leq k$. We view an initial segment of the natural numbers as the corridor. Then p_x^t gets a familiar meaning: $\mathcal{M}, i \models p_x^t$ means that "Tile T_t is placed at position (x, i).

Having this interpretation it seems straightforward to write down the constraints:

- The side edges are tiled by T_0:

$$p_0^0 \wedge p_{n-1}^0$$

- Every position is tiled by exactly one tile:

$$\bigwedge_{0 \leq x \leq n-1} p_x^0 \vee \ldots \vee p_x^k.$$

$$\bigwedge_{0 \leq x \leq n-1,\ 0 \leq t \neq t' \leq k} \neg(p_x^t \wedge p_x^{t'}).$$

- Colors match horizontally by

$$\bigwedge_{0 \leq x < n-1,\ 0 \leq t, t' \leq k} ((p_x^t \rightarrow \bigvee_{\{t' | \mathrm{right}(T_t) = \mathrm{left}(T_{t'})\}} p_{x+1}^{t'}).$$

- Colors match vertically by

$$\bigwedge_{0 \leq x \leq n-1,\ 0 \leq t, t' \leq k} ((p_x^t \rightarrow \Box \bigvee_{\{t' | \mathrm{up}(T_t) = \mathrm{down}(T_{t'})\}} p_x^{t'}).$$

- Every row either has a successor or it is the last one:

$$\Diamond \top \vee (p_0^0 \wedge \ldots \wedge p_{n-1}^0).$$

These are all the universal constraints. Now we need worlds in which the formula $(p_0^0 \wedge \ldots \wedge p_{n-1}^0)$ is true, one for the bottom row and one for the top. It is natural to use the locally satisfiable formula for the bottom row. Thus let

$$A ::= (p_0^0 \wedge \ldots \wedge p_{n-1}^0) \wedge \Diamond \top.$$

$\Diamond\top$ is there to ensure that the real tiling starts. But how are we to ensure that the corridor ends? It is conceivable that A is satisfiable in an infinite model making all constraints true in which never another world containing $(p_0^0 \wedge \ldots \wedge p_{n-1}^0)$ exists. This encoding seems hopeless but a closer inspection of the problem helps us out.

Suppose an instance is a yes instance. Then there exists some m such that $n \times m$ can be tiled. We have placed no maximum on m but a little combinatorics show that we can. If a yes instance exists, there exists one in which no row is repeated in the corridor. (Because if there is, we can cut the whole intermediate part out and still have a yes instance.) How many rows of length n can we make with $k+1$ tiles? At most $(k+1)^n$. That gives an upper bound on m. If a tiling exists there exists one with at most $1+(k+1)^n$ many rows. Now we can use the counter formula from Section 2.5 to create a binary counter starting at 0 at the world in which A is true and we just add a constraint stating that the world with counter value $(k+1)^n$ is the last but one world in the model. The counter formula has size polynomial in k and n and the complete formula is polynomial in the size of the input. We leave the details of checking correctness to the reader. ❑

Corridor tiling is highly suited for linear structures but it does not seem easy to use it for establishing the PSPACE lower bound for local **K** satisfiability. For one thing we cannot enforce such deep models without constraints. On the other hand, we can enforce models with an exponential number of leaves. Luckily there are also two person game versions of square and corridor tiling, which are PSPACE and EXPTIME complete, respectively. One play of a game is naturally represented by a sequence of game-positions, representing the moves made by the two players. All plays on a specific board can then be represented by a tree, such that each play is a complete branch in the tree. (For instance, think of chess. Each node represents the board at that stage of the game. The root represents the start position. It has exactly as many successors as there are possible opening moves for white. From each successor there is again one successor for each legal move of black, and so on.) When working with games we are often not interested if a player can win *some* play, rather we want to know if she can win *every* play. If so, we say she has a winning strategy.

As modal logic and trees are closely related, it seems a good idea to encode game trees. That is just what we will do. Tiling games are played as follows. There are two players, a male and a female who alternate in placing tiles. She starts at the origin and they work their way up from left to right, and at the end of a row they start again at the left most position of the next row. Players must obey the matching color rules. She wins a play is she can establish a complete tiling. Otherwise he wins. She has a winning strategy is she can win every play. Clearly she can win *some* play iff a tiling exists. But the question whether she can win *every* play seems harder. And indeed

FACT 24.

1. The problem whether she has a winning strategy in the square tiling game (n given in unary) is PSPACE complete.
2. The same problem for corridor tiling is EXPTIME complete.

We use the square tiling game to show PSPACE hardness of local **K** satisfiability. The corridor game can be used to show EXPTIME hardness for the satisfiability problem under constraints. This is a fairly straightforward extension of the square game, using the same type of variables as in corridor tiling and also using a counter formula. Complete proofs are given in [6, 37, 4].

PROPOSITION 25. *The local satisfiability problem for the basic modal system is PSPACE hard.*

Proof. Let $T = \{T_0, \ldots, T_k\}$, n be an instance of the square tiling game. Use the same propositional variables as in square tiling. Use the branch and store macros of Section 2.1 to create an at most $n^2 - 1$ deep tree containing all possible legal plays. (Add the required extra constraints about matching colors and the uniqueness of tiles from square tiling where needed.) It might be convenient to use an extra variable encoding whose turn it is. Let that be she indicating that she is to move. Now there exists a tiling if there is an $n^2 - 1$ deep path in the tree. The female player has a winning strategy if she can win every game. We can formalize this using the notion of a winning position. Let wp be a variable denoting whether a position is winning for the female player. All leaves at depth $n^2 - 1$ are winning, and only those leaves. Now we inductively define winning positions for the intermediate nodes in the tree. If she is to move, then a position is winning iff she can move to a winning position:

$$she \rightarrow (wp \leftrightarrow \Diamond wp).$$

If he is to move, then a position is winning for her iff he can move at all, and all his moves lead to winning positions for her:

$$\neg she \rightarrow (wp \leftrightarrow (\Diamond \top \wedge \Box wp).$$

Of course these formulas must be true everywhere in the game tree. But, because the tree is of depth only $n^2 - 1$, this can be enforced by a formula of polynomial size in n. Now she has a winning strategy iff the root of the game tree is a winning position for her. Further details are left to the reader. ❑

To end this section we give a more involved reduction, based on ideas from Lewis [26]. As it is the most difficult one, we spell it out completely. We show NEXPTIME hardness for a number of modal systems of which the two-dimensional modal logic from Section 2.7 is a special case. All have the same language and the models are of the same shape, with two accessibility relations H and V. Let Grid be the class of models satisfying

(17) $\forall xy(\exists z(xVz \wedge zHy) \leftrightarrow \exists z(xHz \wedge zVy))$

(18) $\forall xyz((xHy \wedge xVz) \rightarrow \exists w(yVw \wedge zHw))$.

These frame conditions make the models have a grid like nature. (17) states that $V - H$ and $H - V$ paths commute, while (18) states a Church–Rosser like condition. They are both Sahlqvist definable by respectively $\Diamond_h \Diamond_v \varphi \leftrightarrow \Diamond_v \Diamond_h \varphi$ and $\Diamond_v \Box_h \varphi \rightarrow \Box_h \Diamond_v \varphi$, respectively. These conditions play an important role in the study of products of modal logic [15]. Note that all two dimensional models are in Grid.

PROPOSITION 26. *Let* Grid$'$ *be any subclass of* Grid *containing all finite two dimensional frames. The local satisfiability problem for the modal system with class of models* Grid$'$ *is NEXPTIME hard.*

Proof. Let Grid$'$ be as in the proposition. We will reduce the square tiling problem with n given in binary to the satisfiability problem of the modal system with class of models Grid$'$. Without loss we may assume that we have to tile the $2^n \times 2^n$ square. Let an instance $(2^n, T)$, with $T = \{T_0, \ldots, T_t\}$ be fixed.

We will define a formula $A_{n,T}$ which describes this instance. In order to obtain the lower bound we show that

(A) $A_{n,T}$ is computable in polynomial time in n and $|T|$.

(B) if T tiles $2^n \times 2^n$, then $A_{n,T}$ is satisfiable in a finite two dimensional model, and

(C) if $A_{n,T}$ is Grid'–satisfiable, then T tiles $2^n \times 2^n$.

Given (A), (B) and (C) we have an effective reduction from the NEXPTIME-complete tiling problem to Grid'-satisfiability, and the result follows.

We first describe how we represent a T-tiling of $2^n \times 2^n$ in an $\mathbf{S5}^2$ model. Let (N, E) be a binary tree of depth $2n$ with nodes N and edges E. The elements of N are $2n$-long strings in the alphabet $\{0, 1, \Box\}$, placed in the tree as in Figure 2. (We use the symbol $\Box$ to make all nodes a string of the same length).

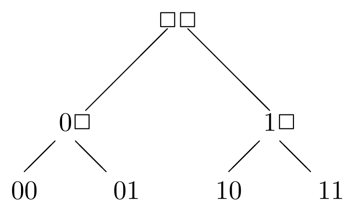


Figure 2. binary tree of depth 2

Let $\mathcal{F} = (N \times N, \equiv_0, \equiv_1)$ be the two dimensional frame with base N. Our formula $A_{n,T}$ will contain the following propositional variables:

d	denotes the nodes in the tree, located on the diagonal
e	denotes the daughter relation in the tree
$p_1, \ldots, p_{2n}$	p_i stands for the i-th bit of the strings encoding the nodes
$t_0, \ldots, t_t$	one variable for each tile,

and variables $p_1^h \ldots p_{2n}^h, t_0^h \ldots t_t^h, p_1^v \ldots p_{2n}^v, t_0^v \ldots t_t^v$, whose meaning will become clear later on.

We now describe a valuation v on $\mathcal{F}$ of these variables. (Because we use V for one of the accessibility relations, we denote the valuation of a model by v in this proof.)

$$\begin{array}{lcl}
\mathsf{v}(d) & = & \{(x, y) \in N^2 \mid x = y\} \\
\mathsf{v}(e) & = & \{(x, y) \in N^2 \mid xEy\} \\
\mathsf{v}(p_i) & = & \{(x, x) \in N^2 \mid x(i) = 1\} \\
\mathsf{v}(p_i^v) & = & \{(x, y) \in N^2 \mid (x, x) \in \mathsf{v}(p_i)\} \\
\mathsf{v}(p_i^h) & = & \{(x, y) \in N^2 \mid (y, y) \in \mathsf{v}(p_i)\}.
\end{array}$$

Let T be a tiling of the $2^n \times 2^n$-grid. A pair $(k, l) \in 2^n \times 2^n$ can be represented by a binary number of length $2n$; the first n places for k and the second n for l. We now make a tile variable t_i true at a leaf x in the tree (N, E) precisely if the tile T_i tiles the pair (k, l) whose representation is x, that is

$$\begin{array}{lcl}
(x, x) \in \mathsf{v}(t_i) & \iff & T_i \text{ tiles } (k, l) \text{ and } (\forall i \leq n) x(i) = k(i) \text{ and} \\
& & \qquad (\forall n < i \leq 2n) x(i) = l(i - n).
\end{array}$$

The t_i^h and t_i^v variables obtain their valuation just as the p_i^h and p_i^v.

We now describe $A_{n,T}$ and show that it is satisfied at the pair (a, a) in $(\mathcal{F}, \mathsf{v})$, where a is the root of the tree (N, E).

We first describe a binary tree of depth $2n$, using the propositional variables $p_1, \ldots p_{2n}$. This will provide us with $(2^n)^2$ leaves each encoding an element in the $2^n \times 2^n$ grid. We use here the "branch and store" formula from Section 2.1 again.

Let $\langle \mathsf{E} \rangle A$ be an abbreviation for $\Diamond_1(e \wedge \Diamond_0(d \wedge A))$, and define $[\mathsf{E}]A = \neg\langle \mathsf{E} \rangle \neg A$. We use $\langle \mathsf{E} \rangle$ as an ordinary **K**-modality. $[\mathsf{E}]^n$ is an abbreviation defined as: $[\mathsf{E}]^0 A = A$ and $[\mathsf{E}]^{n+1} A = [\mathsf{E}][\mathsf{E}]^n A$.

$$d \wedge [\mathsf{E}]^{2n}[\mathsf{E}]\bot \textstyle\bigwedge_{k<2n} [\mathsf{E}]^k \, [\, (\langle \mathsf{E} \rangle p_{k+1} \wedge \langle \mathsf{E} \rangle \neg p_{k+1}) \;\wedge$$

$$\textstyle\bigwedge_{i \le k}((p_i \to [\mathsf{E}]p_i) \wedge (\neg p_i \to [\mathsf{E}]\neg p_i)) \,] \,. \tag{19}$$

Note that the leaves in such a tree make $d \wedge [\mathsf{E}]\bot$ true. Clearly (19) is satisfied at (a, a). The other formulas have a rather redundant formulation if we think about two-dimensional models. Since we want to prove the proposition for a wide class of logics, we have to use this particular formulation. The next two formulas say that on any leaf, precisely one tile variable t holds. Clearly they are satisfied at (a, a).

$$\Box_0^{2n}\Box_1^{2n}[(d \wedge [\mathsf{E}]\bot) \quad \to \quad \bigvee_{1 \le i \le t} t_i] \tag{20}$$

$$\Box_0^{2n}\Box_1^{2n}[\bigwedge_{1 \le i \le t} (t_i \quad \to \quad \bigwedge_{j \ne i} \neg t_j)]. \tag{21}$$

The following formula ensures that the tile T_0 is placed along the edges.

$$\Box_0^{2n}\Box_1^{2n}[d \wedge [\mathsf{E}]\bot \wedge$$

$$((\neg p_1 \wedge \ldots \wedge \neg p_n) \vee (p_1 \wedge \ldots \wedge p_n) \vee (\neg p_{n+1} \wedge \ldots \wedge \neg p_{2n}) \vee \tag{22}$$

$$(p_{n+1} \wedge \ldots \wedge p_{2n})) \to t_0]$$

The next set of formulas capture the behavior of the variables indexed by h and v. First we write formulas which take care of the proper inheritance of information: Let x_i stand for any of $\{p_i, t_i, \neg p_i, \neg t_i\}$

$$\Box_1^{2n}[\Diamond_0^{2n}(x_i) \to x_i^h] \tag{23}$$

$$\Box_0^{2n}[\Diamond_1^{2n}(x_i) \to x_i^v] \tag{24}$$

Then we propagate the new variables in the right direction. Obviously $(\mathcal{F}, \mathsf{v}) \models$ (23)–(26).

$$\Box_1^{2n}[x_i^h \to \Box_0^{2n} x_i^h] \tag{25}$$

$$\Box_0^{2n}[x_i^v \to \Box_1^{2n} x_i^v] \tag{26}$$

Now we can express that colors match:

$$\begin{aligned}\Box_0^{2n}\Box_1^{2n}\, [\, x_h = x_v \wedge y_h = y_v + 1 \wedge t_i^v \wedge \bigvee\{t_j^h \mid 1 \le j \le t\} \to \\ \bigvee\{t_j^h \mid \text{up}(T_i) = \text{down}(T_j)\}\,]\end{aligned} \tag{27}$$

$$\begin{aligned}\Box_0^{2n}\Box_1^{2n}\, [\, y_h = y_v \wedge x_h = x_v + 1 \wedge t_i^v \wedge \bigvee\{t_j^h \mid 1 \le j \le t\} \to \\ \bigvee\{t_j^h \mid \text{right}(T_i) = \text{left}(T_j)\}\,]\,.\end{aligned} \tag{28}$$

Here we use the following abbreviations:

$x_h = x_v$	abbreviates	$\bigwedge_{i \le n}(p_i^h \leftrightarrow p_i^v)$
$x_h = x_v + 1$	abbreviates	$\bigvee_{i \le n}\, [\bigwedge_{l<i}(p_l^h \leftrightarrow p_l^v) \wedge p_i^h \wedge \neg p_i^v \wedge \bigwedge_{i<l \le n}(\neg p_l^h \wedge p_l^v)],$

and similar for the y coordinates where we use the p_i's between p_{n+1} and p_{2n}.

We show that (27) holds everywhere in the model $(\mathcal{F}, \mathsf{v})$. Let $(k,l) \models x_h = x_v \wedge y_h = y_v{+}1 \wedge t_i^v \wedge \bigvee\{t_j^h \mid 1 \leq j \leq t\}$. Then both k and l are leaves in (N,E) by the valuation of the t_i. Then k encodes a pair (x,y) and l a pair $(x,y{+}1)$ in the grid. Because $(k,l) \models t_i^v$, $(k,k) \models t_i$, and T_i tiles (x,y). But since colors match $(x,y{+}1)$ must be tiled by a T_j such that $top(T_i) = down(T_j)$. Then $(l,l) \models t_j$ and indeed $(k,l) \models t_j^h$, as desired.

So we have provided the formula $A_{n,T}$ and shown (B). Clearly the length of this formula is polynomial in $|T|$ and n, and it can effectively be obtained given n and T. This proves (A). Finally we show that the given formula is indeed powerful enough to describe a tiling. So let $A_{n,T}$ be Grid' satisfiable. Then $A_{n,T}$ is satisfied in a model $\mathcal{M} = (W,H,V,\mathsf{v})$ which satisfies (17) and (18). Now suppose that $\mathcal{M}, w \models A_{n,T}$. Then formula (19) forces a binary tree of depth $2n$ starting at w, in which the leaves encode all possible valuations of the p_i-variables. By (17), every leaf in that tree can be reached from w by making $2n$ horizontal, followed by $2n$ vertical steps. So (20) and (21) ensure that at each leaf precisely one tile variable holds. Choose such a tree starting at w and define a tiling of the grid, using the encoding given above. By (20) and (21) this is a well-defined tiling. By (22), the tile T_0 is placed along the edges. Now we check that colors match. Suppose that T_i tiles (x,y) and T_j tiles $(x,y+1)$. Then by (17) we have the following situation in our model:

$wV^{2n}aH^{2n}l$ and $wH^{2n}bV^{2n}k$ for some worlds a and b,

where k is a leaf encoding (x,y) and $\mathcal{M},k \models t_i$ and l a leaf encoding $(x,y{+}1)$ and $\mathcal{M},l \models t_j$. By the inheritance formulas (23) and (24) we have $\mathcal{M},a \models (x,y{+}1)^h \wedge t_j^h$ and $\mathcal{M},b \models (x,y)^v \wedge t_i^v$, where $(x,y{+}1)^h$ abbreviates the conjunction of p_i^h and $\neg p_i^h$ such that p_i and $\neg p_i$ are true at l, etcetera. Now by (18) we have a world c such that $aH^{2n}c$ and $bV^{2n}c$, so $\mathcal{M},c \models (x,y+1)^h \wedge t_j^h \wedge (x,y)^v \wedge t_i^v$ by the propagation formulas (25) and (26). But then by (27) the colors must indeed match. The same argument with (28) shows that colors match horizontally. This proves (C), hence the proposition. ❑

3.6 Language design and complexity

We briefly look at the effect of the design of the modal language on the complexity of the satisfiability problem. We restrict ourselves to the basic modal language. Modal languages all come with an infinite number of propositional variables. What happens if we fix them to some arbitrary finite number n? That is, all formulas are build from propositional variables $p_1, \ldots, p_n$ only. Let us first look at propositional logic. For the full language the satisfiability problem is NP hard. Note that we crucially used the fact that we have an unbounded number of proposition letters in the lower bound proof. If we restrict the number of variables to some fixed n, the problem becomes solvable in linear time: write out the truth table, it contains at most 2^n rows, and make all (at most 2^n) checks. Each check can be done using linear time in the length of the input wff. Of course we must still make a lot of checks but the number is fixed by the design of the language, and not dependent on the input.

Does this also hold for the satisfiability problem of the basic modal system? Halpern [19] showed that it does not. With one propositional variable the satisfiability problem is still PSPACE hard. Here we will explain the idea and give a more efficient encoding as well. What we need is a polynomial time computable translation from arbitrary modal formulas to modal formulas in just one propositional variable, say q, which preserves satisfiability. Then PSPACE hardness of

the full modal language is inherited by the small language. This is Halpern's translation:

$$(p_i)^t \quad = \quad \Diamond(\neg q \wedge \Diamond^i q)$$
$(\cdot)^t$ commutes with all connectives

Again $\Diamond^i$ abbreviates a string of i diamonds, thus the translation is only linear if the i in p_i is given in unary. If i is given in binary, the translation is exponential. We now show that for each wff A, A is satisfiable iff A^t is satisfiable. The right to left direction is easy and left to the reader. For the other direction, let $\mathcal{M} = (W, R, V)$ and let $\mathcal{M}, w_0 \models A$. Create a new model $\mathcal{M}' = (W', R', V')$ such that W' consists of W plus a suitable supply of new worlds, $R'_{\upharpoonright W} = R$, $W \subseteq V'(q)$ and for all $w \in W$ add a path $wRw_1R \ldots Rw_j$ to $\mathcal{M}'$ iff $\mathcal{M}, w \models p_j$. The w_i are all different and not yet in W. Of these new worlds only set $w_j \in V'(q)$. Use new worlds for every $w \in W$ and every propositional variable p_j. An induction on the length of the formula shows that for all $w \in W$, for all subformulas B of A, $\mathcal{M}, w \models B$ iff $\mathcal{M}', w \models B^t$. As $\mathcal{M}, w \models A$ that means that A^t is satisfied in $\mathcal{M}$.

This idea can be used to get a linear translation if the indices of the p's are given in binary. Instead of coding p_i by an i long string of worlds, we encode it by a string of worlds which mimicks the binary encoding of i. For instance, the encoding of p_{101} would be

$$\Diamond(\neg q \wedge \Diamond(q \wedge \Diamond(\neg q \wedge \Diamond q))),$$

in which the leading $\neg q$ is as before to separate "coding" worlds from "real" worlds and the rest of the world sequence is a replica of the string 101. The length of the string of worlds is fixed for all p_i and depends only on the number of variables in the input formula to be translated. Clearly the translation is linear. Details are left to the reader.

Note that we need formulas of unbounded modal depth to encode as many propositional variables as we like. What if we fix the modal depth of our formulas? Then the problem becomes NP complete which is an immediate consequence of the tree model property. Recall that each satisfiable formula A is satisfiable in a tree of depth d the modal depth of A and branching factor b bounded by the number of diamond formulas in A, hence by $|A|$. The number of worlds in such a tree is bounded by b^{d+1}, which —since d is now considered fixed— is simply linear in $|A|$.

If we fix both the number of propositional variables and the modal depth we are rapidly leaving the realm of logic, as then there are only a finite number of wffs up to logical equivalence. A straightforward argument shows that the satisfiability problem then can be decided in linear time. See [19] for further details.

4 HISTORICAL NOTES

We give pointers to further literature and try to sketch a little of the historical development by mentioning some of the milestones. This is not intended to be exhaustive. The chapters 4, 13 and 17 on computational modal logic, description logic and automata for temporal logic, respectively contain useful pointers as well. A detailed set of historical notes together with references to the relavant literature can be found in the notes ending Chapter 6 of [4].

For background, one can consult textbooks on computational complexity [27, 32] and finite model theory [10, 28]. The chapter on computability and complexity in [4] contains detailed arguments of most proofs which are just sketched here and an extended annotated bibliography.

One might say that Cook's Theorem bootstrapped the field of complexity of modal logics [9]. Ladner [25] established the first completeness results for the best known modal logics, focusing on local satisfiability: NP–completeness for S5 and PSPACE–completeness for K,T and S4. The K-World algorithm is from that paper, but using the abstract tableaux developed by Hemaspaandra [36]. The branch and store formula from section 2.1 originates also from [25]. The field took off with Fisher–Ladner [14] and Pratt [33], showing the EXPTIME lower and upper bound for Propositional Dynamic Logic PDL, respectively. This was a large generalization from earlier work: PDL is a multi modal language and the definition of $\models$ is second-order. Moreover within PDL the satisfiability problem under constraints can be reduced to the local satisfiability problem. Pratt introduced the technique of elimination of Hintikka Sets.

In the eighties the field exploded: results for ever more complex modal systems were obtained. Halpern and Moses [20] (the original is from 1985) researched epistemic logic with multiple modalities and transitive closure operators in the tradition of Ladner and Pratt. PSPACE completeness of linear temporal logic with until was settled by Sistla and Clarke [35]. Vardi and Wolper [41] showed how Büchi tree automata can be used for powerful modal logics. A little later, the complexity of the satisfiability problem of one of the most expressive modal logics — the modal μ-calculus— was settled (still EXPTIME–complete) by Emerson and Jutla [11]. More recently, Vardi and co-authors have shown that this result is robust under a number of expansions of the language [40, 34, 24].

There are a number of motivations for studying the complexity of modal satisfiability problems. Description logic is concerned with building knowledge bases and the main computational question is whether a knowledge base is consistent. This field contains a wealth of complexity results, for an enormous variety of modal systems, cf., the handbook [3]. Temporal logic model checking is used in automated program verification, see [7, 8]. Epistemic logic and combinations of epistemic and temporal logics play a role in the analysis of distributed computer programs. Cf., [13] and numerous articles by the authors of that book.

Several attempts have been made to explain why so many modal logics are decidable and why so often their satisfiability problem under constraints is complete for EXPTIME. Vardi [39] emphasizes the tree model property, Andréka, van Benthem, Németi [2] the fact that $\models$ is defined in a guarded fragment of first (or second) order logic, which are really two sides of the same coin, as shown by Grädel [17]. This explanation focuses on the definition of $\models$ and works if the class of structures of a modal system is large enough to allow tree models. Hemaspaandra [36] looks at the frame conditions and their influence, and also at the effect of joining several modal systems. Marx, Venema [30] indicate the importance of "locality" principles (like being in a fixed variable fragment) and the "looseness" principle given by the tree model property. The use of tiling to establish lower bounds has a number of —usually very enthusiastic— advocates, cf., for instance [37, 27, 5, 21].

ACKNOWLEDGEMENT

Some of the material in this chapter has been taken from [30]. The author thanks Martin Otto for very helpful comments.

BIBLIOGRAPHY

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.

[3] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *Description Logic Handbook*. Cambridge University Press, 2002.

[4] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[5] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Springer Verlag, 1997.

[6] B. Chlebus. Domino-tiling games. *J. Comput. Syst. Sci.*, 32:374–392, 1986.

[7] E. Clarke, O. Grumberg, and D. Peled. *Model checking*. MIT Press, 1999.

[8] E.M. Clarke and B.-H. Schlingloff. Model checking. pages 1367–1522. Elsevier Science Publishers, 2000.

[9] S. Cook. The complexity of theorem proving procedures. In *Proc. 3rd ACM SYMP. on Theory of Computing*, pages 151–158, 1971.

[10] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Perspectives in Mathematical Logic. Springer, 1995.

[11] E. Emerson and C. Jutla. The complexity of tree automata and logics of programs. In *Proceedings of the 29th IEEE Symposium on Foundations of Computer Science*, pages 328–337. IEEE Computer Society Press, 1988.

[12] R. Fagin. *Contributions to the Model Theory of Finite Structures*. PhD thesis, U. C. Berkeley, 1973.

[13] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge*. The MIT Press, 1995.

[14] M. Fisher and R. Ladner. Propositional dynamic logic of regular programs. *J. Comput. Syst. Sci.*, 18(2):194–211, 1979.

[15] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*. Number 148 in Studies in Logic and the Foundations of Mathematics. Elsevier, North-Holland, 2003.

[16] M.R. Garey and D.S. Johnson. *Computers and Intractibility. A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[17] E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64(4):1719–1742, 1999.

[18] E. Grädel, P. Kolaitis, and M. Vardi. On the decision problem for two–variable first order logic. *Bulletin of Symbolic Logic*, 3(1):53–69, 1997.

[19] J. Halpern. The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic. *Artificiall Intelligence*, 75(2):361–372, 1995.

[20] J.Y. Halpern and Y.O. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.

[21] D. Harel. Recurring dominoes: making the highly undecidable highly understandable. In *Proc. of the Conference on Foundations of Computing Theory*, volume 158 of *LNCS*, pages 177–194. Springer, 1983.

[22] D. Harel. Recurring dominoes: making the highly undecidable highly understandable. *Annals of Discrete Mathematics*, 24:51–72, 1985.

[23] N. Immerman. *Descriptive Complexity*. Springer, 1998.

[24] O. Kupferman, U. Sattler, and M. Vardi. The complexity of the graded mu-calculus. In *Proceedings of the Conference on Automated Deduction*, volume 2392 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, 2002.

[25] R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM journal of computing*, 6(3):467–480, 1977.

[26] H. Lewis. Complexity results for classes of quantificational formulas. *Journal of Computer and System Sciences*, 21:317–353, 1980.

[27] H.R. Lewis and C.H. Papadimitriou. *Elements of the Theory of Computation (second edition)*. Prentice-Hall, 1998.

[28] L. Libkin. *Elements of Finite Model Theory*. Springer, 2004.

[29] M. Marx and Y. Venema. *Multi-dimensional Modal Logic*. Applied Logic Series. Kluwer Academic Publishers, 1997.

[30] M. Marx and Y. Venema. Local variations on a loose theme: modal logic and decidability. In E. Grädel, P. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Vardi, Y. Venema, and S. Weinstein, editors, *Finite Model Theory and its Applications*. Springer–Verlag, 2004.

[31] M. Mortimer. On languages with two variables. *Zeitsch. f. math. Logik und Grundlagen d. Math.*, 21:135–140, 1975.

[32] Ch. Papadimitriou. *Computational Complexity*. Addison–Wesley, 1994.

[33] V. Pratt. Models of program logics. In *Proceedings of the 20th IEEE symposium on Foundations of Computer Science*, pages 115–122, 1979.

[34] U. Sattler and M. Vardi. The hybrid mu-calculus. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proceedings of the International Joint Conference on Automated Reasoning*, volume 2083 of *LNAI*, pages 76–91. Springer Verlag, 2001.

[35] A. Sistla and E. Clarke. Complexity of propositional linear temporal logics. *J. Assoc. Comput. Mach.*, 32(3):733–749, 1985.

[36] E. Spaan. *Complexity of modal logics*. PhD thesis, University of Amsterdam, Institute for Logic, Language and Computation, 1993.

[37] P. van Emde Boas. The convenience of tilings. In A. Sorbi, editor, *Complexity, Logic and Recursion Theory*, volume 187 of *Lecture notes in pure and applied mathematics*, pages 331–363. Marcel Dekker Inc., 1997.

[38] M. Vardi. On the complexity of bounded–variable queries. In *Proceedings of the ACM 14th Symposium on Principles of Database Systems*, pages 266–276, 1995.

[39] M. Vardi. Why is modal logic so robustly decidable? In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science 31*, pages 149–184. American Math. Society, 1997.

[40] M. Vardi. Reasoning about the past with two-way automata. *Lecture Notes in Computer Science*, 1443:628+, 1998.

[41] M.Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *J. Comput. Syst. Sci.*, 32:183–221, 1986.

[42] Y. Venema. Cylindric modal logic. *Journal of Symbolic Logic*, 60(2):591–623, 1995.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

4

COMPUTATIONAL MODAL LOGIC

Ian Horrocks, Ullrich Hustadt, Ulrike Sattler, and Renate Schmidt

1 Introduction . . . 181
2 Syntax, semantics, and reasoning problems of modal logics . . . 183
3 Translation-based methods . . . 186
3.1 Local satisfiability in multi modal $\mathbf{K}_n$ . . . 186
3.2 Global satisfiability, non-logical axioms, transitive modalities, and $\mathbf{K4}_n$ . . . 200
3.3 Converse modalities and the modal logics $\mathbf{KB}_n$ and $\mathbf{KB4}_n$ . . . 202
3.4 Implementation and optimisation . . . 203
3.5 Other extensions (counting, nominals) . . . 207
4 Tableau-based algorithms . . . 207
4.1 Tableau algorithms in general . . . 208
4.2 Local satisfiability for multi modal $\mathbf{K}_n$ . . . 209
4.3 Transitive modalities and $\mathbf{K4}_n$ . . . 217
4.4 Non-logical axioms and background theories . . . 219
4.5 Converse modalities . . . 222
4.6 Converse modalities and background theories . . . 223
4.7 Other extensions (counting, nominals, transitive closure, and fixpoints) . . . 224
5 Other computational approaches . . . 226
5.1 Automata-based algorithms . . . 226
5.2 Modal resolution . . . 227
5.3 Sequent-based approaches . . . 229
5.4 Inverse method . . . 230
6 Other reasoning problems . . . 231
6.1 Model checking . . . 231
6.2 Proof checking . . . 231
6.3 Computing correspondences . . . 232
6.4 Model generation . . . 233
6.5 Bisimulation . . . 234
6.6 Modal logic programming . . . 236
7 Review and Discussion . . . 237

1 INTRODUCTION

As we have seen in preceding chapters, the worst case complexity of basic reasoning tasks, such as deciding the satisfiability of a modal formula, is at least NP-complete for almost all modal logics. Moreover, for logics extended with features that are useful in practice, the worst case complexity can be much higher, e.g., ExpTime-complete for $\mathbf{K}_n$ extended with non-logical

axioms (background theories), and NExpTime-complete for $\mathbf{K}_n$ extended with converse modalities, graded modalities and nominals.

Some may regard these results as discouraging and the question arises whether automated computation with such logics can be feasible in practice. Fortunately, the kinds of pathological formulae/theories that give rise to these worst case results seem to be rarely encountered in realistic applications, and this has allowed for the successful development and deployment of automated reasoning systems for modal logics and their notational variant, description logics; see Chapter 13 of this handbook. Applications of such systems include, e.g., multi-agent systems [53, 60, 196], configuration [137], conceptual modelling [73], information integration [32], and ontology tools and applications [125, 131, 167, 189, 138, 197].

Even for application derived formulae/theories, however, naive implementations of theoretical proof systems, such as the tableau calculi presented in Chapter 2 of this handbook, are unlikely to be of practical utility. As pointed out in [40], without the use of an analytic cut rule, the minimal length of proofs using these calculi can exceed that of proofs using the truth table method for certain propositional (and modal) formulae. Further, not only is it important that short proofs exist, but also how we go about finding a proof or a counter-model. Much of the work presented in this chapter deals with techniques that reduce the size of the search space or help to traverse the search space more efficiently. Successful modern reasoning systems crucially employ specialised reasoning techniques along with optimisations to dramatically improve typical case performance; cf. for example [85, 90, 97, 115, 116, 158, 159]. In this chapter, we focus on reasoning and optimisation techniques used in tableau-based algorithms and translation-based methods.

Translation-based methods make use of the fact that a wide variety of modal logics can be translated into first-order logic; in fact, they can be considered as characterising certain fragments of first-order logic as explained in Section 2 of Chapter 1 of this handbook. To the translated modal formulae, we can apply first-order reasoning methods, in particular, refinements of resolution [16]. Using this combination of a translation method and resolution has some obvious advantages. Any modal logic which can be embedded into first-order logic can be treated. The translations are straightforward, and can be performed in time $O(n \log n)$, so the engineering effort is minimal. For the resolution part, standard resolution provers can be used, or otherwise they can be used with small adaptations. Modern resolution provers [169, 183, 194] are among the most sophisticated and fastest first-order logic theorem provers currently available. The translation method is generic, it can handle first-order modal logics, undecidable modal logics, and combinations of modal and non-modal logics. In all cases, soundness and completeness of the method is immediate from results showing that the translation is satisfiability equivalence preserving and the soundness and completeness of the resolution calculus for first-order logic. The semi-decidability of first-order logic and the behaviour of first-order resolution on first-order formulae does not give us, however, any immediate insight into the modal fragment of first-order logic, which certainly is decidable, or the behaviour of first-order resolution on translated modal formulae. While termination of a resolution derivation from a translated modal formula is not always guaranteed, there are various ways, using different translations and different refinements of resolution, of obtaining translation-based decision procedures. In Section 3, we discuss some of these approaches and illustrate them using the modal logics $\mathbf{K}_n$, $\mathbf{K4}_n$, $\mathbf{KB}_n$, $\mathbf{K}_n^{\smile}$ ($\mathbf{K}_n$ with converse modalities), and $\mathbf{KB4}_n$. Also, using the modal logic $\mathbf{K}_n$, we want to provide some fundamental understanding of how modern resolution provers work in general, what kind of optimisations are available, and how they can be used to provide effective and practical decision procedures for modal logics.

Tableau-based algorithms are closely related to the prefixed tableau systems presented in Section 6 of Chapter 2 of this handbook. In Section 4, we first explain the exact relationship between the two before describing a tableau algorithm which decides the satisfiability of formulae in the basic multi-modal logic $\mathbf{K}_n$. We then discuss implementation and optimisation techniques which can be used to turn this tableau algorithm into an effective and practical decision procedure for $\mathbf{K}_n$. Following the same structure, we also describe tableau-based algorithms for the modal logics $\mathbf{K4}_n$, $\mathbf{K}_n$ with non-logical axioms, $\mathbf{K}_n^{\smile}$, and their combinations and discuss implementation issues of those algorithms. Whereas the $\mathbf{K}_n$ tableau algorithm terminates "automatically", we use certain cycle detection mechanisms to ensure termination for other modal logics. It can be easily seen that these mechanisms must be chosen carefully to preserve correctness of the algorithm and, at the same time, to enable termination as soon as possible so as to avoid an unnecessarily long search. Interestingly, it has been shown by state of the art description logic reasoners [159, 90, 160] that such tableau algorithms are amenable to optimisation, and that they behave better than their worst-case complexity or that of the corresponding reasoning problem suggest: they implement non-deterministic double exponential decision procedures for logics that are ExpTime-complete.

In Section 5, we give an overview of alternative computational approaches to the satisfiability problem in modal logics. These include automata-based algorithms, direct resolution, the inverse method, and sequent-based approaches. In Section 6, we survey reasoning problems other than satisfiability and provability which are relevant for applications of modal logics, namely, model checking, proof checking, and computing correspondence properties for modal axiom schemata. Finally, we conclude the chapter with a brief review and discussion of current and future research.

2 SYNTAX, SEMANTICS, AND REASONING PROBLEMS OF MODAL LOGICS

Throughout this chapter, we use a notation that is compatible with the one presented in Chapter 1 of this handbook. We will use the symbols $p, q, p_i, q_i, \ldots$ for propositional variables. Here, we will be concerned with extensions and variants of the multi-modal logic $\mathbf{K}_n$. The set of $\mathbf{K}_n$ *formulae* is the smallest set that contains all propositional variables, is closed under Boolean operators, and contains $[i]\psi$ and $\langle i\rangle\psi$ for each $1 \leq i \leq n$ and each $\mathbf{K}_n$ formula ψ. Formulae of the form $[i]\psi$ and $\langle i\rangle\psi$ are called *box formulae* and *diamond formulae*, respectively. In different sections, we will consider different normal forms of $\mathbf{K}_n$ formulae, and thus we are generous here and allow all kinds of Boolean operators and abbreviations, e.g. $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\top$ (for any tautology), $\perp$ (for $\neg\top$), etc.

As usual, the semantics of $\mathbf{K}_n$ is defined in terms of relational, Kripke structures or frames. A *frame* is a tuple $\langle W, R\rangle$ of a non-empty set W (of worlds) and a mapping R from natural numbers i, $1 \leq i \leq n$ to binary relations over W, thus $R(i) \subseteq W \times W$. Here and in the rest of the chapter, we use R_i as an abbreviation of $R(i)$, and we say that w is *i-accessible from v* if $R_i(v, w)$. A *model* is given by a triple $\mathfrak{M} = \langle W, R, V\rangle$, where $\langle W, R\rangle$ is a frame and V is a mapping from propositional variables to subsets of W. The notion of a formula ψ *being true* in a model $\mathfrak{M}$ at a world $w \in W$ is inductively defined as follows (we omit the definition for most Boolean operators).

$$
\begin{array}{llll}
\mathfrak{M}, w & \models & p & \text{iff } w \in V(p) \\
\mathfrak{M}, w & \models & \neg\psi & \text{iff not } \mathfrak{M}, w \models \psi \\
\mathfrak{M}, w & \models & \psi \wedge \phi & \text{iff } \mathfrak{M}, w \models \psi \text{ and } \mathfrak{M}, w \models \phi \\
\mathfrak{M}, w & \models & \psi \vee \phi & \text{iff } \mathfrak{M}, w \models \psi \text{ or } \mathfrak{M}, w \models \phi \\
\mathfrak{M}, w & \models & [i]\psi & \text{iff } \mathfrak{M}, v \models \psi, \text{ for all } v \text{ with } R_i(w, v) \\
\mathfrak{M}, w & \models & \langle i \rangle\psi & \text{iff } \mathfrak{M}, v \models \psi, \text{ for some } v \text{ with } R_i(w, v)
\end{array}
$$

A modal formula ϕ is *satisfiable* in $\mathbf{K}_n$ if there exists some $\mathfrak{M} = \langle W, R, V \rangle$ such that, for some $w \in W$, $\mathfrak{M}, w \models \phi$. In this case, we say that ϕ is *satisfied in* $\mathfrak{M}$. ϕ is *valid* in $\mathbf{K}_n$ if, for every $\mathfrak{M} = \langle W, V \rangle$ and every $w \in W$, $\mathfrak{M}, w \models \phi$; ϕ and ψ are *equivalent* if $\phi \leftrightarrow \psi$ is valid.

As usual, satisfiability and validity are inter-reducible, i.e., ϕ is satisfiable iff $\neg\phi$ is *not* valid, and ϕ is valid iff $\neg\phi$ is *un*satisfiable. Thus, in what follows, we will mostly concentrate on a single inference problem, namely satisfiability testing. It is well-known that the satisfiability problem (and thus validity) in $\mathbf{K}_n$ is PSpace-complete [93, 129] (see also Chapter 3 of this handbook), and there are various decision procedures for this problem [49, 93, 129] and implementations thereof [87, 90, 120, 159]. Many of these procedures exploit the fact that any satisfiable $\mathbf{K}_n$ formula is satisfied in a finite tree model (i.e., one where the relational structure of the frame forms a finite tree) of depth linear in the size of the input formula. In this chapter, we will discuss in depth a resolution-based algorithm (in Section 3) and a tableau-based algorithm (in Section 4) for the satisfiability of $\mathbf{K}_n$, and then explain how these two basic algorithms can be modified to also decide more expressive modal logics.

We will often restrict our attention to formulae in *negation normal form (NNF)*. In formulae in NNF, $\wedge$, $\vee$, and $\neg$ are the only Boolean connectives used, and negation occurs only in front of propositional variables. Each formula of $\mathbf{K}_n$ and all extensions of $\mathbf{K}_n$ discussed in this chapter can be easily transformed into an equivalent formula in NNF in linear time, by pushing negation inwards, using a combination of de Morgan's laws and the duality between box and diamond formulae.

In this chapter we refer to a number of extensions of $\mathbf{K}_n$ which we define in the following.

$\mathbf{K4}_n$, $\mathbf{KB}_n$, and $\mathbf{KB4}_n$. We will discuss decision procedures for $\mathbf{K4}_n$, the multi-modal logic of *transitive frames*, $\mathbf{KB}_n$, the multi-modal logic of *symmetric frames*, and $\mathbf{KB4}_n$, the multi-modal logic of *symmetric and transitive frames*. All these logics share the same language with $\mathbf{K}_n$, but their semantics is based on different classes of frames. As $\mathbf{K4}_n$ models, we only consider those models that are based on frames (W, R) in which each R_i is *transitive*, i.e., where for any $u, v, w \in W$, $R_i(u, v)$ and $R_i(v, w)$ imply $R_i(u, w)$. For example, $\langle i \rangle\langle i \rangle p \wedge [i]\neg p$ is $\mathbf{K}_n$ satisfiable, whereas the same formula is $\mathbf{K4}_n$ unsatisfiable. As $\mathbf{KB}_n$ models, we only consider models that are based on frames (W, R) in which each R_i is *symmetric*, i.e., where for any $u, v \in W$, $R_i(u, v)$ implies $R_i(v, u)$. An example of a formula which is $\mathbf{K}_n$ and $\mathbf{K4}_n$ satisfiable, but $\mathbf{KB}_n$ unsatisfiable, is $\neg p \wedge \langle i \rangle [i] p$. Finally, $\mathbf{KB4}_n$ models are based on frames (W, R) where each R_i is symmetric and transitive. The formula $\langle i \rangle \neg p \wedge \langle i \rangle [i] p$, for example, is $\mathbf{KB4}_n$ unsatisfiable, but satisfiable in $\mathbf{K}_n$, $\mathbf{K4}_n$, and $\mathbf{KB}_n$.

The modal logic $\mathbf{K4}_n$ is axiomatised by the axioms of the modal logic $\mathbf{K}_n$ (see, for example, Chapter 2 of this handbook), plus the axiom schema $\mathbf{4}$ below. Similarly, $\mathbf{KB}_n$ is axiomatised by adding the axiom schema $\mathbf{B}$ to the axiomatisation of $\mathbf{K}_n$. Finally, for $\mathbf{KB4}_n$, we add both axiom schemas $\mathbf{B}$ and $\mathbf{4}$.

Axiom $\mathbf{4}$ $\qquad [i]\phi \to [i][i]\phi$

Axiom $\mathbf{B}$ $\qquad \phi \to [i]\langle i \rangle \phi$

The modal logic $\mathbf{K4}_n$ is of interest since both tableau-based algorithms and translation-based methods require additional techniques to ensure termination. The modal logic $\mathbf{KB}_n$ only requires a rather straightforward modification of the reasoning procedures we present for $\mathbf{K}_n$, but raises some implementation issues for tableau systems. It is also worthwhile to remember that, in classical tableau systems, the treatment of $\mathbf{KB}_n$ requires some form of *cut rule*. Finally, the procedures we present for $\mathbf{KB4}_n$ combine the techniques we introduce for $\mathbf{K4}_n$ and $\mathbf{KB}_n$.

Non-logical axioms. We consider *background theories*, i.e., finite sets $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ of *non-logical axioms* γ_i. A model $\mathfrak{M}$ *satisfies* a background theory Γ if, for each $w \in W$ and each $\gamma \in \Gamma$, $\mathfrak{M}, w \models \gamma$. As a reasoning problem, we are interested in the satisfiability of a formula ϕ w.r.t. a background theory Γ, i.e., whether there exists a model $\mathfrak{M}$ that satisfies both ϕ and a background theory Γ. Note that, in such a model, ϕ has to be true in at least one world, whereas all formulae in Γ have to be true in all worlds.

Next, we explain why we discuss algorithms that reason w.r.t. background theories. Firstly, considering background theories makes reasoning more difficult, i.e., satisfiability w.r.t. background theories is ExpTime-complete [173], and thus they present a considerable challenge for automated reasoning tools. As with $\mathbf{K4}_n$, tableau algorithms for reasoning w.r.t. background theories no longer terminate on all inputs. However, in contrast to $\mathbf{K4}_n$, background theories allow us to enforce models with paths of exponential length using the standard encoding of incrementation modulo 2^n on n propositional variables representing a binary counter (see, for example, page 14 of [133] for such a formula). The latter can be viewed as a symptom of the increased complexity since we might have to consider and search models with paths of exponential length. Secondly, background theories can be viewed as a weak form of the *universal modality*, i.e., ϕ is satisfiable w.r.t. Γ iff

$$\phi \wedge \bigwedge_{\gamma \in \Gamma} [0]\gamma,$$

is satisfied in a model based on a frame $\langle W, R\rangle$ with $R_0 = W \times W$. In such a model, $[0]$ is called the *universal modality* because it can be used to access all worlds. In [4], it was shown how to reduce satisfiability in $\mathbf{K}_n$ with the universal modality (i.e., where the universal modality might also occur at a deeper modal level) to satisfiability w.r.t. background theories. Thirdly, background theories can be used to "internalise" axioms or "circumscribe" frame conditions. To do this, we first restrict our attention to formulae in negation formal form. As an example, let us consider $\mathbf{K4}_n$ as discussed above and a $\mathbf{K4}_n$ formula ϕ in NNF. It can be shown that ϕ is satisfiable iff ϕ is $\mathbf{K}_n$ satisfiable w.r.t. the background theory

$$\{[i]\psi \rightarrow [i][i]\psi \mid [i]\psi \text{ is a subformula of } \phi \text{ and } 1 \leq i \leq n\}.$$

Finally, background theories are notational variants of description logic *TBoxes* or *terminologies*, which are used in applications to hold the intensional domain knowledge [44, 173]; see also Chapter 13 of this handbook. A TBox is (the description logic variant of) a background theory of the form $\{\phi_i \rightarrow \psi_i \mid 1 \leq i \leq m\}$. Restricting non-logical axioms to implications is (i) not a real restriction: we can transform each background theory Γ into a single implication of the form $\top \rightarrow \bigwedge_{\psi \in \Gamma} \psi$; (ii) quite natural in various application, and (iii) enables the use of efficient optimisation techniques in tableau systems [109], which we will discuss in Section 4.4.

Converse modalities. We discuss modifications of our algorithms to decide satisfiability of $\mathbf{K}_n^{\smile}$ formulae w.r.t. background theories, where $\mathbf{K}_n^{\smile}$ is the extension of $\mathbf{K}_n$ that allows the use of *converse* modal parameters $i^{\smile}$ in modalities. That is, $[i^{\smile}]\phi$ and $\langle i^{\smile}\rangle\phi$ are also well-formed formulae. The mapping R is extended to converse modal parameters as follows: $R_{i^{\smile}} = \{(w, v) \mid$

$R_i(v, w)\}$. Converse modalities are of interest because they occur naturally in applications, e.g., in description logics [105, 171] and temporal logics [163, 187], and because they require reasoning techniques that are able to reason in both directions over relations. For example, to detect the unsatisfiability of the formula $q \wedge \langle i \rangle (p \wedge [i^{\smile}]\neg q)$, one has to reason both ways over R_i. This is similar to the kind of reasoning required for $\mathbf{KB}_n$ but slightly more tricky since, in $\mathbf{K}_n^{\smile}$, reasoning in "both ways" over a relation depends also on the worlds related by R_i. For example, the $\mathbf{KB}_n$ formula $[i]\psi$ is equivalent to the $\mathbf{K}_n^{\smile}$ formula $[i]\psi \wedge [i^{\smile}]\psi$.

Graded/deterministic modalities. We discuss modifications of the tableau algorithm to handle deterministic and graded modalities. The former have (atomic) modal parameters i whose interpretation R_i has to be a functional relation. To understand graded modalities, note that a diamond formula $\langle i \rangle \phi$ can be read as "in at least one i-related world, ϕ is true", and a box formula $[i]\phi$ can be read as "in at most zero i-related worlds, ϕ is not true". *Graded* modalities generalise these formulae: $\mathbf{K}_n^c$ (resp. $\mathbf{K}_n^{\smile,c}$) is the extension of $\mathbf{K}_n$ (resp. $\mathbf{K}_n^{\smile}$) where we also allow for formulae of the form $\langle i \rangle_m \phi$ and $[i]_m \phi$. We read $\langle i \rangle_m \phi$ as "in at least $(m+1)$ i-related worlds, ϕ is true" and $[i]_m \phi$ as "in at most m i-related worlds, ϕ is true". The semantics is extended in the obvious way.

$\mathfrak{M}, w \models \langle i \rangle_m \phi$ iff there are at least $m+1$ worlds $v \in W$ with $R_i(w, v)$ and $\mathfrak{M}, v \models \psi$

$\mathfrak{M}, w \models [i]_m \phi$ iff there are at most m worlds $v \in W$ with $R_i(w, v)$ and $\mathfrak{M}, v \models \psi$

Please note that $\neg[i]_m \phi$ is equivalent to $\langle i \rangle_m \neg\phi$ and that, by adding $\top \rightarrow [i]_1 \top$ to our background theory, we restrict models to those in which R_i is a functional relation. Thus algorithms that can handle both graded modalities and background theories can also handle deterministic modalities. From a complexity point of view, adding graded/deterministic modalities rarely effects the worst case complexity, e.g., $\mathbf{K}_n^c$ and $\mathbf{K}_n^{\smile,c}$ are both PSpace-complete without background theories [93, 186] and ExpTime-complete w.r.t. background theories [173, 184, 186]. From a practical reasoning perspective, graded modalities add quite some difficulty: consider, e.g., the formulae $[i]_3 \top \wedge \langle i \rangle_1 (p \vee q) \wedge \langle i \rangle_1 (\neg p \vee q)$ and $[i]_1 p \wedge [i]_1 \neg p \wedge \langle i \rangle_2 q$. The former is satisfiable, but we have to find that the two diamond formulae can be satisfied via a "common" i-related world. The latter is unsatisfiable, but we have to find that a third i-related world in which neither p nor $\neg p$ holds cannot exist.

Nominals. In their simplest form, nominals [5, 163] are propositional variables that are true in exactly one world; we use $o_1, o_2, \ldots$ for these variables and indicate the availability of nominals in a logic by the superscript $\cdot^o$ as, e.g., in $\mathbf{K}_n^o$. Nominals are of interest for automated reasoning since they destroy the *tree model property (TMP)* (see Chapter 1 of this handbook) of a logic. For example, the formula $o_2 \wedge \langle i \rangle (o_1 \wedge \langle i \rangle o_2)$ has only models with a cycle of length two. We mentioned above that $\mathbf{K}_n$ enjoys this property (and so do its extensions with converse and graded modalities and background theories), and that this property is exploited by tableau- and some resolution-based algorithms. Interestingly, adding nominals to $\mathbf{K}_n^{\smile}$ takes the complexity from PSpace- to ExpTime-completeness [5], and adding nominals to $\mathbf{K}_n^{\smile,c}$ takes the complexity to NExpTime-completeness [186]. Here, we will consider $\mathbf{K}_n^{\smile,o}$ with background theories.

3 TRANSLATION-BASED METHODS

3.1 *Local satisfiability in multi modal* $\mathbf{K}_n$

As outlined in Chapter 1 of this handbook, using the *standard translation* formulae of the basic modal logic $\mathbf{K}$ can be embedded into first-order logic. This translation is also called the *relational translation*, since it is based on the relational Kripke semantics for modal logic. In the following

we denote this translation by π_r and present here its straightforward generalisation to multi-modal $\mathbf{K}_n$.

$$\begin{aligned}
\pi_r(\top, x) &= \top & \pi_r(\bot, x) &= \bot \\
\pi_r(p, x) &= P_p(x) & \pi_r(\neg\varphi, x) &= \neg\pi_r(\varphi, x) \\
\pi_r(\varphi \star \psi, x) &= \pi_r(\varphi, x) \star \pi_r(\psi, x) \quad \text{for } \star \in \{\wedge, \vee, \rightarrow, \leftrightarrow\} \\
\pi_r([i]\varphi, x) &= \forall y\,(R_i(x,y) \rightarrow \pi_r(\varphi, y)) & \pi_r(\langle i\rangle\varphi, x) &= \exists y\,(R_i(x,y) \wedge \pi_r(\varphi, y))
\end{aligned}$$

In the translation, each propositional variable p is uniquely associated with a unary predicate symbol P_p, while each modal parameter i, $1 \leq i \leq n$, is uniquely associated with a binary predicate symbol R_i. In addition, x is an arbitrary first-order variable while y is an arbitrary first-order variable distinct from x.

This translation is *satisfiability equivalence preserving*, that is, for every modal formula φ, φ is $\mathbf{K}_n$ satisfiable iff $\pi_r(\varphi, x)$ is first-order satisfiable. The free variable x is assumed to be existentially quantified.

The currently predominant method for reasoning about first-order formulae is *resolution* [170]. The method requires that a first-order formula or set of first-order formulae φ is first transformed into a satisfiability equivalent set of clauses N_0. This set of clauses is then *saturated* using the *resolution* rule and *factoring* rule shown in Figure 1. That is, given a clause set N_i, $i \geq 0$, these inference rules are applied (top-down) to clauses already in the set and the conclusion C of such an application is added to N_i to give us the clause set N_{i+1}. This process continues until either (i) the current clause set N_i contains the empty clause (i.e. $\bot$) or (ii) no new clauses can be derived, that is, any conclusion of an application of the resolution and factoring rules to clauses in N_i is already contained in N_i. Any clause set containing the empty clause is unsatisfiable. Thus, in the case (i), N_i is unsatisfiable and by the soundness of the resolution calculus, so is N_0. This implies that φ is unsatisfiable, since N_0 is satisfiable iff φ is satisfiable. In the case (ii), if N_i does not contain the empty clause, N_i is satisfiable and it is possible to construct a model for N_i. By the completeness of the resolution calculus, N_0 is satisfiable. It follows that φ is satisfiable. Due to the undecidability of first-order logic, there is in general no guarantee that, after a finite number of steps, we either always encounter case (i) or (ii). If we apply the inference rules of the resolution calculus in a *fair* way, then the completeness of the resolution calculus ensures that, eventually, the empty clause is derived. However, if the formula φ and the clause set N_0 are satisfiable, then the saturation process may continue indefinitely (unless suitable resolution refinements and/or translation methods are used, see below).

The last observation is also true if the formula φ we are considering belongs to a decidable fragment of first-order logic or is the result of translating a formula belonging to a decidable modal logic like $\mathbf{K}_n$. However, starting with [123], a large number of fragments of first-order logic have been shown to be decidable by resolution or refinements of resolution [48, 65, 76, 82, 117, 179].

Resolution: $\dfrac{C \vee A_1 \quad \neg A_2 \vee D}{(C \vee D)\sigma}$
where σ is the most general unifier of atoms A_1 and A_2

Factoring: $\dfrac{C \vee L_1 \vee L_2}{(C \vee L_1)\sigma}$
where σ is the most general unifier of literals L_1 and L_2

Figure 1. The resolution calculus R

Let us start by considering what happens if we apply the basic unrefined resolution calculus to the relational translation of modal formulae. Consider, for example, the modal formula $\varphi_1 = [2](p \to \langle 1 \rangle p)$. Its translation $\pi_r(\varphi_1, x)$ is the first-order formula $\psi_1^r = \forall y\, (R_2(x, y) \to (P_p(y) \to \exists z\, (R_1(y, z) \wedge P_p(z))))$. The corresponding set of clauses N_0^r consists of the two clauses

$$
\begin{array}{ll}
(1) & \neg R_2(a, x) \vee \neg P_p(x) \vee R_1(x, f(x)) \\
(2) & \neg R_2(a, y) \vee \neg P_p(y) \vee P_p(f(y))
\end{array}
$$

where a is a constant introduced during the clause form transformation (for the free variable x in ψ_1^r). We assume that the variables in two clauses to which we want to apply the resolution rule are renamed so that they are variable disjoint, and we consider such *variant clauses* to be equal. There are several possibilities to apply the resolution rule to clauses (1) and (2). For example, we can resolve clause (1) on its second literal, $\neg P_p(x)$, with clause (2) on its third literal, $P_p(f(z))$. The conclusion is

$$
\text{[(1)2,R,(2)3]} \quad (3) \quad \neg R_2(a, f(z)) \vee R_1(f(z), f(f(z))) \vee \neg R_2(a, z) \vee \neg P_p(z)
$$

Clause (3) resolves with clause (2) and yields

$$
\begin{array}{lll}
\text{[(2)3,R,(3)4]} & (4) & \neg R_2(a, f(y)) \vee \neg R_2(a, f(f(y))) \vee R_1(f(f(y)), f(f(f(y)))) \\
& & \qquad \vee \neg R_2(a, y) \vee \neg P_p(y)
\end{array}
$$

This clause also resolves with clause (2), and again, the conclusion resolves with (2), and so forth. Repeatedly resolving the newly derived clauses with (2) yields clauses with increasingly more literals and increasingly more complex terms. All these clauses are new, that is, none is the same as an input clause or a clause derived earlier. As the formula φ_1 and its translation ψ_1^r are satisfiable, we are not able to derive the empty clause. So, the saturation process will continue indefinitely.

There are three approaches that we can take to solve this termination problem:

1. We can develop and use *alternative translations* of modal formulae to first-order (clause) logic, and try to find a translation for which resolution is a decision procedure.

2. We can develop and use *refinements of resolution* which restrict the application of the inference rules of the resolution calculus and use powerful redundancy elimination methods.

3. We can develop and use *alternative inference methods* for first-order logic.

These three approaches are not mutually exclusive, in particular, alternative translations can be combined with both refinements of resolution and alternative calculi.

Investigations following the first approach have resulted in the introduction of the *optimised functional translation* of $\mathbf{K}_n$ to sorted first-order logic, more precisely, to a monadic fragment of sorted first-order logic called *basic path logic* [149, 177]. Basic path logic has a sort S_W for the set of worlds W and a sort S_i for each modal parameter i, $1 \leq i \leq n$, in a modal logic. It has n binary functions $[\_\,\_]_i$ of sort $S_W \times S_i \to S_W$. Also there are special unary predicates def_i of sort S_W representing subsets of W. Each propositional variable p is uniquely associated with a unary predicate symbols P_p of sort S_W. Commonly, the optimised functional translation π_{of} is defined as a two step process: (i) the application of the functional translation to a modal formula which translates it to basic path logic, followed by (ii) the application of a quantifier exchange operation which converts the first-order formula obtained from the functional translation into

prenex normal form and moves all existential quantifiers inwards as far as possible. Since we focus here only on the satisfiability problem, we can give a simplified presentation of the optimised functional translation obtained in just one step.

$$\pi_{of}(\top, s) = \top \qquad \pi_{of}(\bot, s) = \bot$$
$$\pi_{of}(p, s) = P_p(s) \qquad \pi_{of}(\neg\varphi, s) = \neg\pi_{of}(\varphi, s)$$
$$\pi_{of}(\varphi \star \psi, s) = \pi_{of}(\varphi, s) \star \pi_{of}(\psi, s) \quad \text{for } \star \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$$
$$\pi_{of}([i]\varphi, s) = \forall y{:}S_i(\mathit{def}_i(s) \rightarrow \pi_{of}(\varphi, [s\, y{:}S_i]_i))$$
$$\pi_{of}(\langle i\rangle\varphi, s) = \mathit{def}_i(s) \wedge \pi_{of}(\varphi, [s\, y{:}S_i]_i)$$

where s denotes a *(world) path* and $y{:}S_i$ denotes a variable of sort S_i. The omission of the quantifiers in the definition for $\langle i\rangle\varphi$ is intensional. The optimised functional translation of a modal formula φ in negation normal form is now given by $\pi_{of}(\varphi, x{:}S_W)$, where $x{:}S_W$ is an arbitrary variable of sort S_W, and $x{:}S_W$ as well as the $y{:}S_i$ from $\pi_{of}(\langle i\rangle\varphi, s)$ are free variables which are implicitly existentially quantified.

As an example, consider again the modal formula $\varphi_1 = [2](p \rightarrow \langle 1\rangle p)$. Its optimised functional translation is $\varphi_1^{of} = \pi_{of}(\varphi_1, x{:}S_W) = \forall y{:}S_2\,(\mathit{def}_2(x{:}S_W) \rightarrow (P_p([x{:}S_W\, y{:}S_2]_2) \rightarrow (\mathit{def}_1([x{:}S_W\, y{:}S_2]_2) \wedge P_p([[x{:}S_W\, y{:}S_2]_2\, z{:}S_1]_1))))$.

In the representation of paths we often remove all occurrences of the binary functions $[\_\,\_]_i$ except for the outermost occurrence and also leave out the index of that remaining occurrence, e.g. $[[x{:}S_W\, y{:}S_2]_2\, z{:}S_1]_1$ is written as $[x{:}S_W\, y{:}S_2\, z{:}S_1]$. It is straightforward to restore the original path based on the remaining information. Intuitively, a path term like $[x{:}S_W\, y{:}S_2\, z{:}S_1]$ represents a path from a world x to another, possible identical, world in a Kripke frame via a series of 'steps' along the accessibility relations of the frame. Here an R_2-step is followed by an R_1-step, which is indicated by the sorts S_2 and S_1 associated with the variables y and z, respectively. The def_i predicates express 'definability' for a world in the sense that $\mathit{def}_i(s)$ is true iff the world s has an i-successor.

The standard translation π_r accommodates axiom schemas like $\mathbf{4}$ and $\mathbf{B}$ by adding first-order formulae

$$(R_{\mathbf{B}}) \qquad \forall x\, y\,(R_i(x, y) \rightarrow R_i(y, x))$$
$$(R_{\mathbf{4}}) \qquad \forall x\, y\, z\,((R_i(x, y) \wedge R_i(y, z)) \rightarrow R_i(x, z))$$

representing the *relational frame properties* corresponding to these axiom schemas. By contrast, in the case of the optimised functional translation, we add so-called *functional frame properties* in the form of (conditional) equations between path terms, for example

$$(F_{\mathbf{B}}) \qquad \forall x{:}S_W \forall y{:}S_i \exists z{:}S_i\,(\mathit{def}_i(x{:}S_W) \rightarrow \mathit{def}_i([x{:}S_W\, y{:}S_i]) \wedge \mathit{def}_i(x{:}S_W) \rightarrow x{:}S_W = [[x{:}S_W\, y{:}S_i]\, z{:}S_i])$$
$$(F_{\mathbf{4}}) \qquad \forall x{:}S_W \forall y{:}S_i \forall z{:}S_i \exists u : S_i\,((\mathit{def}_i(x{:}S_W) \wedge \mathit{def}_i([x{:}S_W\, y{:}S_i])) \rightarrow [[x{:}S_W\, y{:}S_i]\, z{:}S_i] = [x{:}S_W\, u : S_i])$$

Functional frame properties corresponding to axiom schemas $\mathbf{5}$, $\mathbf{D}$, $\mathbf{T}$, $\mathbf{G}$, as well as functional frame properties corresponding to weak density, irreflexivity, and McKinsey's axiom can be found in [177].

THEOREM 1 ([149]). *Let $\mathbf{K}_n\Sigma$ be a complete modal logic such that the functional frame properties corresponding to the axiom schemas in Σ are a set of first-order formulae F_Σ. Then φ is satisfiable in $\mathbf{K}_n\Sigma$ iff $F_\Sigma \wedge \pi_{of}(\varphi, x{:}S_W)$ is first-order satisfiable.*

If we are only interested in establishing the satisfiability of formulae in the basic modal logic $\mathbf{K}_n$ or extensions of $\mathbf{K}_n$ by the axiom schema $\mathbf{D}$ for some or all modalities, then the use of sorted first-order logic and binary function symbols can be avoided by using k-ary predicates where the sort information is coded into the predicate names [95, 116]. The k-ary predicate symbols are $P_{p,\sigma}$ and $def_{i,\sigma}$ where p denotes a propositional symbol, σ is a k-sequence of natural numbers and $1 \leq i \leq n$, $n \geq 0$. We use $\overline{x}$ to denote a sequence of variables $x_1, \ldots, x_k$, and we use 'ϵ' and '.' for the empty sequence and the concatenation operation on sequences, respectively.

$$\pi'_{of}(\top, \overline{x}, k, \sigma) = \top \qquad\qquad \pi'_{of}(\bot, \overline{x}, k, \sigma) = \bot$$

$$\pi'_{of}(p, \overline{x}, k, \sigma) = \begin{cases} P_{p,\epsilon} & \text{if } \sigma = \epsilon \text{ and } k = 0 \\ P_{p,\sigma}(x_1, \ldots, x_k) & \text{otherwise} \end{cases}$$

$$\pi'_{of}(\neg\varphi, \overline{x}, k, \sigma) = \neg\pi'_{of}(\varphi, \overline{x}, k, \sigma)$$

$$\pi'_{of}(\varphi \star \psi, \overline{x}, k, \sigma) = \pi'_{of}(\varphi, \overline{x}, k, \sigma) \star \pi'_{of}(\psi, \overline{x}, k, \sigma) \quad \text{for } \star \in \{\wedge, \vee\}$$

$$\pi'_{of}([i]\varphi, \overline{x}, k, \sigma) = \forall x_{n+1}\, (def_{i,\sigma}(\overline{x}) \rightarrow \pi'_{of}(\varphi, \overline{x}.x_{k+1}, k{+}1, \sigma.i))$$

$$\pi'_{of}(\langle i\rangle\varphi, \overline{x}, k, \sigma) = def_{i,\sigma}(\overline{x}) \wedge \pi'_{of}(\varphi, \overline{x}.x_{k+1}, k{+}1, \sigma.i)$$

The translation of a modal formula φ in negation normal form is given by $\pi'_{of}(\varphi, \epsilon, 0, \epsilon)$. In the case of the modal logic $\mathbf{KD}_n$, and in fact for any modal logic where an accessibility relation R_i is serial, all occurrences of $def_{i,\sigma}$ can be replaced with the logical constant $\top$.

The translation π'_{of}, called the polyadic optimised functional translation, takes advantage of two observations:

1. All paths in $\pi_{of}(\varphi, x{:}S_W)$ start with $x{:}S_W$ and since this variable is free, it is implicitly existentially quantified, and is interpreted as a constant. Therefore, removing $x{:}S_W$ from all paths is a satisfiability equivalence preserving transformation.

2. The variables in the paths occurring in $\pi_{of}(\varphi, x{:}S_W)$ are *prefix stable*, that is, for any variable $x_{i+1}{:}S_{j_i}$ there exists a unique prefix $[x{:}S_W\, x_0{:}S_{j_0}\, \ldots\, x_i{:}S_{j_i}]$ such that every path containing $x_{i+1}{:}S_{j_i}$ has the form $[x{:}S_W x_0{:}S_{j_0} \ldots x_i{:}S_{j_i} x_{i+1}{:}S_{j_{i+1}} \ldots x_k{:}S_{j_k}]$. Thus, if a variable occurs at position i in one path, then it occurs at position i in all paths. This property is due to a characteristic ordering of variables in the path terms determined by the structure of modal formulae and is a reflection of the tree model property. Also, since variables do not 'move', the sort information can be associated with the position at which a variable occurs instead of with the variable itself. Thus, we can code the sort information into predicate names. Replacing $P_p([x{:}S_W\, x_0{:}S_{j_0}\, \ldots\, x_k{:}S_{j_k}])$ by $P_{p,S_W S_{j_0} \ldots S_{j_k}}(x, x_0, \ldots, x_k)$ is therefore a satisfiability equivalence preserving transformation.

Taking both observations together, and also taking advantage of the assumption that each sort S_i is uniquely identified by its index i, we see that we can replace $P_p([x{:}S_W\, x_0{:}S_{j_0}\, \ldots\, x_k{:}S_{j_k}])$ with $P_{p,j_0 \ldots j_k}(x_0, \ldots, x_k)$.

For the modal formula $\varphi_1 = [2](p \rightarrow \langle 1\rangle p)$ the translation using π'_{of} is $\psi_1^{of'} = \forall y\, (def_{2,\epsilon} \rightarrow (P_{p,2}(y) \rightarrow (def_{1,2}(y) \wedge P_{p,21}(y, z))))$. The corresponding set $N_0^{of'}$ of clauses again consists of two clauses.

$$\begin{array}{ll} (5) & \neg def_{2,\epsilon} \vee \neg P_{p,2}(y) \vee def_{1,2}(y) \\ (6) & \neg def_{2,\epsilon} \vee \neg P_{p,2}(y) \vee P_{p,21}(y, z) \end{array}$$

Unlike for the clausal form of the relational translation of φ_1, for these two clauses there is no possibility to apply either the resolution rule or the factoring rule. For, we see that $P_{p,21}(y,z)$ in clause (6) is not unifiable with $P_{p,2}(y)$ in clause (5), nor is it unifiable with $P_{p,2}(y)$ in clause (6). So, for this particular example we are able to conclude that $N_0^{of'}$, $\psi_1^{of'}$, and φ_1 are satisfiable without the need to perform a single inference step.

With this translation, is termination of the saturation process guaranteed? Consider the modal formula $\varphi_2 = [2](\neg p \vee [1]q) \vee [2]p$. Its optimised functional translation is

$$\psi_2^{of'} = \forall y(\mathit{def}_{2,\epsilon} \rightarrow (\neg P_{p,2}(y) \vee \forall z(\mathit{def}_{1,2}(y) \rightarrow P_{q,21}(y,z)))) \vee \forall u(\mathit{def}_{2,\epsilon} \rightarrow P_{p,2}(u)).$$

The corresponding set N_0^{of} of clauses consists of just one clause

$$(7) \quad \neg\mathit{def}_{2,\epsilon} \vee \neg P_{p,2}(y) \vee \neg\mathit{def}_{1,2}(y) \vee P_{q,21}(y,z) \vee \neg\mathit{def}_{2,\epsilon} \vee P_{p,2}(u).$$

We consider clauses to be *multisets* of literals, that is, a literal can occur more than once in a clause, as is the case with the literal $\neg\mathit{def}_{2,\epsilon}$ in clause (7). It is sometimes convenient to consider clauses as *sets* of literals. However, this complicates the completeness proof for resolution calculi which commonly proceeds by lifting ground level derivations to the non-ground level. This lifting is easier if clauses are considered to be multisets on both the ground and the non-ground level. Furthermore, multisets make the computational effort explicit which has to go into removing duplicate literals from clauses.

We can resolve clause (7) with itself on the second literal and the last literal. Remember that this means that we first have to generate a variable-disjoint copy of clause (7) to serve as second premise of a resolution step. The conclusion is

$$\begin{aligned}[(7)2,R,(7)6] \quad (8) \quad & \neg\mathit{def}_{2,\epsilon} \vee \neg P_{p,2}(y_1) \vee \neg\mathit{def}_{1,2}(y_1) \vee P_{q,21}(y_1,z_1) \vee \neg\mathit{def}_{1,2}(y_2) \\ & \vee P_{q,21}(y_2,z_2) \vee \neg\mathit{def}_{2,\epsilon} \vee P_{p,2}(u_2) \vee \neg\mathit{def}_{2,\epsilon} \vee \neg\mathit{def}_{2,\epsilon}.\end{aligned}$$

We observe that the number of occurrences of $\neg\mathit{def}_{2,\epsilon}$ has doubled, to four, and also that we now have two subclauses $\neg\mathit{def}_{1,2}(y_1) \vee P_{q,21}(y_1,z_1)$ and $\neg\mathit{def}_{1,2}(y_2) \vee P_{q,21}(y_2,z_2)$ which are variants of each other.[1] Note that these two subclauses are not simply duplicates, so in a clauses-as-sets setting they would still remain, while all the duplicates of $\neg\mathit{def}_{2,\epsilon}$ would not occur. Clause (8) can again be resolved with itself or with clause (7). A possible resolvent is:

$$\begin{aligned}[(7)2,R,(8)8] \quad (9) \quad & \neg\mathit{def}_{2,\epsilon} \vee \neg P_{p,2}(y_3) \vee \neg\mathit{def}_{1,2}(y_3) \vee P_{q,21}(y_3,z_3) \\ & \vee \neg\mathit{def}_{1,2}(y_1) \vee P_{q,21}(y_1,z_1) \vee \neg\mathit{def}_{1,2}(y_2) \vee P_{q,21}(y_2,z_2) \\ & \vee \neg\mathit{def}_{2,\epsilon} \vee P_{p,2}(u_2) \vee \neg\mathit{def}_{2,\epsilon} \vee \neg\mathit{def}_{2,\epsilon} \vee \neg\mathit{def}_{2,\epsilon} \vee \neg\mathit{def}_{2,\epsilon}\end{aligned}$$

We can continue this process indefinitely, producing bigger and bigger clauses. This shows that the saturation process does not terminate.

We observe, however, that resolution is not the only inference rule that can be applied to clause (8): we can also apply the factoring rule. Indeed, there are several possibilities to do so, for example, we can apply the factoring rule to $P_{q,21}(y_1,z_1)$ and $P_{q,21}(y_2,z_2)$. The resulting factor is:

$$\begin{aligned}[(8)4,F,(8)6] \quad (10) \quad & \neg\mathit{def}_{2,\epsilon} \vee \neg P_{p,2}(y_1) \vee \neg\mathit{def}_{1,2}(y_1) \vee \neg\mathit{def}_{1,2}(y_1) \vee P_{q,21}(y_1,z_1) \\ & \vee \neg\mathit{def}_{2,\epsilon} \vee P_{p,2}(u_2) \vee \neg\mathit{def}_{2,\epsilon} \vee \neg\mathit{def}_{2,\epsilon}.\end{aligned}$$

[1] We consider two formulae or clauses to be equal iff they are *variants* of each other, that is, they are syntactically equal modulo variable renaming.

We see that the clause (10) is a subclause of the clause (8) from which it was derived. Clause (10) thus subsumes clause (8). In general, a clause C *subsumes* a clause D iff there is a substitution σ such that $C\sigma$ is a subclause of D. Subsumed clauses are *redundant* and can be removed from a clause set without losing completeness.

Further factoring steps are possible on clause (10), for example, on the two occurrences of $\neg def_{1,2}(y_1)$. It turns out that the final clause that we can derive by a series of factoring steps is a condensation of clause (8). By definition, a *condensation* $\mathrm{Cond}(C)$ of a clause C is a minimal subclause of C which is also an instance of C. A clause C is *condensed* iff there exists no condensation of C which is a strict subclause of C. For any clause C, $\mathrm{Cond}(C)$ subsumes C, and hence C is redundant in the presence of $\mathrm{Cond}(C)$ and can be removed. This justifies that we can systematically replace clauses with their condensation.

The condensation of clauses (7) and (8) are the clauses

$$\begin{array}{ll} (7') & \neg def_{2,\epsilon} \vee \neg P_{p,2}(y) \vee \neg def_{1,2}(y) \vee P_{q,21}(y,z) \vee P_{p,2}(u) \\ (8') & \neg def_{2,\epsilon} \vee \neg P_{p,2}(y_1) \vee \neg def_{1,2}(y_1) \vee P_{q,21}(y_1,z_1) \vee P_{p,2}(u_2). \end{array}$$

These two clauses are variants of each other, that is, we regard them as equal. So, the only clause derivable from (7′) is identical to (7′), which means the saturation process terminates.

It turns out that systematically replacing clauses by their condensation is sufficient to guarantee termination not only for this particular example formula, but for any modal formula in $\mathbf{K}_n$ or $\mathbf{KD}_n$.

THEOREM 2 ([175]). *Let φ be a modal formula and $N_0 = \pi'_{of}(\varphi, \epsilon, 0, \epsilon)$. Then the saturation process from N_0 by the resolution calculus R defined in Figure 1 in which clauses are systematically and eagerly replaced with their condensation always terminates with a clause set N_n, and φ is $\mathbf{K}(\mathbf{D})_n$ unsatisfiable iff N_n contains the empty clause.*

It is important to understand how Theorem 2 is to be interpreted. The theorem says that R plus condensing is a decision procedure for this translation of modal satisfiability problems. It does not stipulate that we must use R plus condensing. Rather, the theorem sets out the minimal requirement or weakest condition we have to impose on a saturation process by R to ensure that it terminates. It states that, as long as we keep clauses condensed, we can use any refinement of the calculus R, we can perform inference steps on any literals in a clause and can perform inference steps in any order, and the saturation process is still guaranteed to terminate. Condensing can be simulated by factoring and subsumption deletion. Consequently, any first-order theorem prover which implements some refinement of R (and subsumption deletion, which is standardly available) can serve as a decision procedure for $\mathbf{K}_n$ and $\mathbf{KD}_n$.

The question as to which particular refinement of resolution to use (determining which inference steps are required for completeness) and which particular strategies and heuristics to use (determining the order in which inference steps are performed) is then subject to both theoretical and empirical investigation. We leave empirical aspects aside for the moment and instead focus on refinements of resolution.

A wide range of refinements of resolution can be formulated in the general resolution calculus of Bachmair and Ganzinger; full details can be found in [16]. In the general resolution calculus, here denoted by $R^{\succ}_{S}$, inference rules are parameterised by an admissible ordering $\succ$ on literals and a selection function S. Essentially, an admissible ordering is a total (well-founded) strict ordering on the ground level such that for literals: $\ldots \succ \neg A_n \succ A_n \succ \ldots \succ \neg A_1 \succ A_1$. This is extended to the non-ground level in a canonical manner. A *selection* function S assigns to each clause a possibly empty set of occurrences of negative literals. If C is a clause, then the

Deduction: $$\frac{N}{N \cup \{\mathrm{Cond}(C)\}}$$
if C is either a resolvent or a factor of clauses in N.

Deletion: $$\frac{N \cup \{C\}}{N}$$
if C is redundant in N.

Splitting: $$\frac{N \cup \{C \vee D\}}{N \cup \{C\} \mid N \cup \{D\}}$$
if C and D are variable-disjoint.

Resolution: $$\frac{C \vee A_1 \qquad \neg A_2 \vee D}{(C \vee D)\sigma}$$
where (i) σ is the most general unifier of atoms A_1 and A_2, (ii) no literal is selected in C, and $A_1\sigma$ is strictly $\succ$-maximal with respect to $C\sigma$, and (iii) $\neg A_2$ is either selected, or $\neg A_2\sigma$ is maximal with respect to $D\sigma$ and no literal is selected in D.

Positive Factoring: $$\frac{C \vee A_1 \vee A_2}{(C \vee A_1)\sigma}$$
where (i) σ is the most general unifier of atoms A_1 and A_2, and (ii) no literal is selected in C and $A_1\sigma$ is $\succ$-maximal with respect to $C\sigma$.

Figure 2. Expansion and inference rules of $R^{\succ}_{S}$

literal occurrences in $S(C)$ are *selected*. No restrictions are imposed on the selection function. The calculus comprises *expansion rules* of the general form

$$\frac{N}{N_1 \mid \cdots \mid N_n}$$

where both the numerator N and the denominators $N_1, \ldots, N_n$ ($n \geq 1$) are finite sets of clauses. Expansion rules are applied top-down. There are three kinds of expansion rules: Deduction, Deletion and Splitting which are defined in Figure 2. The inferences rules consist of the resolution and the factoring rule also defined in Figure 2. The left premise of the resolution rule is called the *positive premise* and the right premise is called the *negative premise*. The implicit assumption is that the premises have no common variables. *Resolvents* are conclusions of resolution steps, while *factors* are conclusions of factoring steps.

A *derivation* in $R^{\succ}_{S}$ from a set of clauses N is a finitely branching, ordered tree T with root N and nodes which are sets of clauses. The tree is constructed by applications of the expansion rules to the leaves. We assume that no resolution or factoring inference (on the same premises) is performed twice on the same branch of the derivation. A branch $N(= N_0), N_1, \ldots$ in a derivation T is called a *closed branch* in T iff the clause set $\bigcup_{j \geq 0} N_j$ contains the empty clause, otherwise it is called an *open branch*. We call a branch B in a derivation tree *complete* (with respect to $R^{\succ}_{S}$) iff no new successor nodes can be added with $R^{\succ}_{S}$ to the endpoint of B, otherwise it is called an *incomplete branch*. A derivation T is a *refutation* iff every path $N(= N_0), N_1, \ldots$ in it is a closed branch, otherwise it is called an *open derivation*.

In general, the calculus $R^{\succ}_{S}$ can be enhanced with standard simplification rules such as tautology deletion and subsumption deletion. In fact, it can be enhanced by all simplification rules which are compatible with a general notion of redundancy [16, 18]. For example, C is redundant in $N \cup \{\mathrm{Cond}(C)\}$. A set N of clauses is *saturated up to redundancy* with respect to a particular refinement of resolution if the conclusion of every inference from non-redundant premises in N is either contained in N, or else is redundant in N. A derivation T from N is called *fair* if, for any path $N(= N_0), N_1, \ldots$ in T with *limit* $N_\infty = \bigcup_{j \geq 0} \bigcap_{k \geq j} N_k$, it is the case that each clause C which can be deduced from non-redundant premises in N_∞ is contained in some N_j. Intuitively, fairness means that no non-redundant inferences are delayed indefinitely. For a finite

complete branch $N(= N_0), N_1, \dots N_n$, the limit N_∞ is equal to N_n.

THEOREM 3 ([18]). *Let T be a fair $\mathcal{R}^{\succ}_{S}$ derivation from a set N of clauses. Then*

1. *if $N(= N_0), N_1, \dots$ is a path with limit N_∞, then N_∞ is saturated (up to redundancy),*
2. *N is satisfiable iff there exists a path in T with limit N_∞ such that N_∞ is satisfiable, and*
3. *N is unsatisfiable iff for every path $N(= N_0), N_1, \dots$ the clause set $\bigcup_{j \geq 0} N_j$ contains the empty clause.*

As an aside, we note that it follows from the decidability result for the optimised functional translation (Theorem 2) that we can use any instance of $\mathcal{R}^{\succ}_{S}$ for the clause sets obtained by applying $\pi_{of'}$ to modal formulae in $\mathbf{K}_n$ or $\mathbf{KD}_n$. In particular, this gives us full flexibility with respect to orderings and selection functions. Furthermore, by Theorem 2, even instances of $\mathcal{R}^{\succ}_{S}$ without splitting will terminate.

The purpose of the ordering $\succ$ and the selection function S is to restrict the set of literals in a clause to which resolution and factoring can be applied. This limits the number of inferences performed and consequently reduces the search space. For example, reconsider the clause set N_0^{of} consisting of just the clause

$$(7) \quad \neg \mathit{def}_{2,\epsilon} \vee \neg P_{p,2}(y) \vee \neg \mathit{def}_{1,2}(y) \vee P_{q,21}(y,z) \vee \neg \mathit{def}_{2,\epsilon} \vee P_{p,2}(u).$$

Using an ordering we could restrict inference steps to the literals $P_{q,21}(y,z)$ and $P_{p,2}(u)$. Now $\neg P_{p,2}(y)$ can no longer be resolved with $P_{p,2}(u)$, since $\neg P_{p,2}(y)$ is neither maximal nor selected. Alternatively, using a selection function we could restrict inference steps to the negative literals $\neg \mathit{def}_{2,\epsilon}$ or $\neg P_{p,2}(y)$. Again, no inference steps are possible.

Let us reconsider our very first example, the set of clauses N_0^r obtained via the relational translation of $\varphi_1 = [2](p \to \langle 1 \rangle p)$:

$$\begin{array}{ll} (1) & \neg R_2(a,x) \vee \neg P_p(x) \vee R_1(x, f(x)) \\ (2) & \neg R_2(a,y) \vee \neg P_p(y) \vee P_p(f(y)) \end{array}$$

If we use an ordering $\succ$ such that $R_1(x, f(x))$ is maximal in clause (1) and $P_p(f(y))$ is maximal in clause (2), then no inference steps are possible on N_0^r. Likewise, if we select the literals $\neg R_2(a,x)$ and $\neg R_2(a,y)$ in their respective clauses, then again no inference steps are possible.

This raises the question whether it is possible to obtain decision procedures for $\mathbf{K}_n$ satisfiability based on the relational translation π_r and the calculus $\mathcal{R}^{\succ}_{S}$ by using particular ordering or selection functions. To simplify matters, we use a technique called structural transformation. The purpose of the structural transformation is to convert the first-order translation into a more manageable form. Before we describe it formally, we need to define some basic notions.

The polarity of (occurrences of) modal or first-order subformulae is defined as usual. Any occurrence of a proper subformula of an equivalence has *zero polarity*. For occurrences of subformulae not below a '$\leftrightarrow$' symbol, an occurrence of a subformula has *positive polarity* if it is inside the scope of an even number of (explicit or implicit) negations, and it has *negative polarity* if it is inside the scope of an odd number of negations. For any first-order formula φ, if λ is the position of a subformula in φ, then $\varphi|_\lambda$ denotes the subformula of φ at position λ and $\varphi[\psi \mapsto \lambda]$ is the result of replacing $\varphi|_\lambda$ at position λ by ψ. The set of all the positions of subformulae of φ is denoted by $\Lambda(\varphi)$.

Structural transformation, also referred to as *renaming*, associates a predicate symbol Q_λ and a literal $Q_\lambda(\overline{x})$ with each element λ of $\Lambda \subseteq \Lambda(\varphi)$, where $\overline{x} = x_1, \dots, x_n$ are the free variables

of $\varphi|_\lambda$, the symbol Q_λ does not occur in φ and two symbols Q_λ and $Q_{\lambda'}$ are equal only if $\varphi|_\lambda$ and $\varphi|_{\lambda'}$ are equivalent formulae. In practice, one may want to use the same symbols for variant subformulae, or subformulae which are obviously equivalent, for example, $\varphi \vee \psi$ and $\psi \vee \varphi$. Let $\mathrm{Def}^+_\lambda(\varphi) = \forall \overline{x}\,(Q_\lambda(\overline{x}) \to \varphi|_\lambda)$ and $\mathrm{Def}^-_\lambda(\varphi) = \forall \overline{x}\,(\varphi|_\lambda \to Q_\lambda(\overline{x}))$. The *definition* of Q_λ is the formula

$$\mathrm{Def}_\lambda(\varphi) = \begin{cases} \mathrm{Def}^+_\lambda(\varphi) & \text{if } \varphi|_\lambda \text{ has positive polarity,} \\ \mathrm{Def}^-_\lambda(\varphi) & \text{if } \varphi|_\lambda \text{ has negative polarity,} \\ \mathrm{Def}^+_\lambda(\varphi) \wedge \mathrm{Def}^-_\lambda(\varphi) & \text{otherwise.} \end{cases}$$

The corresponding clauses are called *definitional clauses*. Now, assume that Λ is a set of positions in a formula φ and that we want to systematically replace subformulae at positions in Λ while adding definitions for the newly introduced predicate symbols. A convenient way to do so, is to start by the renaming innermost subformulae, and then to proceed up towards the root of φ. Formally, define $\mathrm{Def}_\Lambda(\varphi)$ inductively by:

$$\mathrm{Def}_\emptyset(\varphi) = \varphi \quad \text{and} \quad \mathrm{Def}_{\Lambda\cup\{\lambda\}}(\varphi) = \mathrm{Def}_\Lambda(\varphi[Q_\lambda(\overline{x}) \mapsto \lambda]) \wedge \mathrm{Def}_\lambda(\varphi),$$

where λ is maximal in $\Lambda \cup \{\lambda\}$ with respect to the prefix ordering on positions. A *definitional form* of φ is $\mathrm{Def}_\Lambda(\varphi)$, where Λ is a subset of all positions of subformulae of φ (usually, non-atomic or non-literal subformulae).

THEOREM 4 (e.g. [29, 161]). *Let φ be a first-order formula. Then*

1. *φ is satisfiable iff $\mathrm{Def}_\Lambda(\varphi)$ is satisfiable, for any $\Lambda \subseteq \Lambda(\varphi)$, and*
2. *$\mathrm{Def}_\Lambda(\varphi)$ can be computed in polynomial time (or linear time if new symbols are introduced for all formulae occurring in Λ).*

By $\Lambda_m(\varphi)$ we denote the set of positions in $\pi_r(\varphi, x)$ corresponding to non-atomic subexpressions of the modal formula φ.

Structural transformation allows us to keep the structure of the clauses we have to deal with very simple. This in turn simplifies the characterisation of classes of clause sets that can be derived from some initial clause set using $R^{\succ}_S$. For example, assume that, in the relational translation of the modal formula $\varphi_3 = [2]\langle 1\rangle p$, we apply structural transformation to all positions that correspond to non-atomic subexpressions of the original modal formula φ_3. The result is the set of formulae on the left of Figure 3, while the clausal form is given on the right. In general, the formulae we obtain in this way from the relational translation of modal formulae (as well as the corresponding sets of clauses) belong to quite a number of decidable fragments of first-order logic, for example, the two-variable fragment, the guarded fragment [3], Maslov's class K [135], and fluted logic [165, 166]. Resolution decision procedures have been developed for the guarded

$Q_{[2]\langle 1\rangle p}(x)$	$Q_{[2]\langle 1\rangle p}(a)^*$
$\wedge\forall x\,(Q_{[2]\langle 1\rangle p}(x) \to \forall y\,(R_1(x,y) \to Q_{\langle 1\rangle p}(y)))$	$\neg Q_{[2]\langle 1\rangle p}(x) \vee \neg R_2(x,y)^* \vee Q_{\langle 1\rangle p}(y)$
$\wedge\forall x\,(Q_{\langle 1\rangle p}(x) \to \exists y\,(R_1(x,y) \wedge P_p(y)))$	$\neg Q_{\langle 1\rangle p}(x) \vee R_1(x,f(x))^*$
	$\neg Q_{\langle 1\rangle p}(x) \vee P_p(f(x))^*$

Figure 3. The structural transformation and the clausal form of $[2]\langle 1\rangle p$

fragment [48, 76], for Maslov's class K [111, 117], for fluted logic [179] and various other classes related to modal logics, see e.g. [65, 82, 83, 111]. Here we use the results for the clausal class $\mathbf{DL}^*$ defined in [49]. $\mathbf{DL}^*$ is a variation of the class of DL-clauses, that was introduced in [119] for the purpose of deciding expressive description logics.

In order to simplify the definition of $\mathbf{DL}^*$, all clauses are assumed to be maximally split. The components in the variable partition of a clause are called *variable-disjoint* or *split components*, that is, split components do not share variables. If $C_1, \ldots, C_n$ are the split components of C, then we say C can be *decomposed* into $C_1, \ldots, C_n$. A clause which cannot be split further is called a *maximally split clause* or an *indecomposable clause*. Now, a maximally split clause C is a $\mathbf{DL}^*$*-clause* iff the following conditions are satisfied: (i) all literals are unary, or binary; (ii) there is no nesting of function symbols; (iii) every functional term in C contains all the variables of C (this condition implies that, if C contains a functional ground term, then C is ground); (iv) every binary literal (even if it has no functional terms) contains all the variables of C. It can be shown that all clauses in structural form obtained from $\mathrm{Def}_\Lambda(\pi_r(\varphi, x))$ for a modal formula φ belong to $\mathbf{DL}^*$ [49].

In order to decide the class $\mathbf{DL}^*$, we use the following ordering. First we define an order $>_d$ on terms: $s >_d t$ if s is deeper than t, and every variable that occurs in t, occurs deeper in s. Then we define $P(s_1, \ldots, s_n) \succ Q(t_1, \ldots, t_m)$ as $\{s_1, \ldots, s_n\} >_d^{\mathrm{mul}} \{t_1, \ldots, t_m\}$. Here $>_d^{\mathrm{mul}}$ is the multiset extension of $>_d$ [16]. So we have $P(f(x)) \succ P(a), P(x)$ and $P(x, y) \succ Q(x)$, but not $P(f(x)) \succ P(f(a))$. The selection function S is empty. We denote this particular instance of the resolution calculus $\mathsf{R}_S^\succ$ by $\mathsf{R}^{\mathsf{ord}}$.

In the example in Figure 3, the maximal literals (with respect to $\succ$) are marked with *. These are the literals that we may apply resolution or factoring to.

In order to prove that the procedure $\mathsf{R}^{\mathsf{ord}}$ is indeed a decision procedure we have to show that it is complete and terminating. Completeness follows immediately from the completeness of $\mathsf{R}_S^\succ$. Termination follows from the fact, that over a finite signature, there are only finitely many maximally split $\mathbf{DL}^*$-clauses (module variable renaming), and the fact that, from $\mathbf{DL}^*$-clauses, $\mathsf{R}^{\mathsf{ord}}$ produces only clauses that are again in $\mathbf{DL}^*$, or are splittable into components in $\mathbf{DL}^*$ (cf. [111, 119]).

THEOREM 5 ([49, 180]). *Let Σ be an arbitrary set of axiom schemas such that $\mathbf{K}_n\Sigma$ is complete and the clausal form of the relational frame properties F_Σ corresponding to the axiom schemas in Σ are expressible in $\mathbf{DL}^*$. Let φ be a $\mathbf{K}_n$ formula and let N be the clausal form of $F_{\Sigma,\varphi} = F_\Sigma \wedge \mathrm{Def}_{\Lambda_m(\varphi)}(\pi_r(\varphi, x))$. Then*

1. *φ is unsatisfiable in $\mathbf{K}_n\Sigma$ iff $F_{\Sigma,\varphi}$ is first-order unsatisfiable iff there is a refutation of N by $\mathsf{R}^{\mathsf{ord}}$,*
2. *N is a set of $\mathbf{DL}^*$ clauses, and*
3. *any derivation from N in $\mathsf{R}^{\mathsf{ord}}$ (up to redundancy) terminates in double exponential time; if Σ is empty, then any derivation from N in $\mathsf{R}^{\mathsf{ord}}$ (up to redundancy) terminates in exponential time, and*

Here, and in subsequent theorems, we assume that the complexity of redundancy elimination is at most exponential in the size of a clause set. The theorem remains true for $\mathsf{R}^{\mathsf{ord}}$ without the splitting rule, but condensing is key for decidability.

It is usually the case that, when studying modal decidability problems by analysing the decidability of related clausal classes, one comes to realise that stronger results are possible than

initially anticipated. In [49], extensions of $\mathbf{K}_n$ with $\mathbf{PDL}$-like relational operations have been studied. Relational operations expressible in $\mathbf{DL}^*$ include intersection, union, complementation, and converse, as are non-logical axioms.

In R^{ord}, the inferences performed are determined by a refinement based on an ordering and the empty selection function. We now consider results from [49, 119, 121, 180] for a different refinement which is based solely on a selection function and an optional ordering. More precisely, the calculus is based on maximal selection of negative literals. This means the selection function S selects exactly the set of all negative literals in any non-positive clause. When no ordering refinement $\succ$ is used, the resolution rule of $R_S^{\succ}$ can be replaced with the following rule.

Resolution with maximal selection:

$$\frac{C_1 \vee A_1 \quad \cdots \quad C_n \vee A_n \quad \neg A_{n+1} \vee \ldots \vee \neg A_{2n} \vee D}{(C_1 \vee \ldots \vee C_n \vee D)\sigma}$$

provided that for every i, $1 \leq i \leq n$, (i) σ is the most general unifier of A_i and A_{n+i}, (ii) $C_i \vee A_i$ and D are positive clauses, (iii) no A_i occurs in C_i, and (iv) the $\neg A_{n+i}$ are selected. The *negative premise* is $\neg A_{n+1} \vee \ldots \vee \neg A_{2n} \vee D$ and the other premises are the *positive premises*. The literals A_i and A_{n+i} are the *eligible literals*.

Let R^{hyp} be the instance of $R_S^{\succ}$ based on maximal selection and no ordering. This means that the rules are the above resolution rule, positive unordered factoring and splitting. This refinement of resolution is also referred to as *hyperresolution plus splitting*. Condensation is not needed, but could of course be added without losing completeness and will improve the performance of the procedure. Tautology deletion is used as a simplification rule. All derivations in R^{hyp} are generated by strategies in which no application of the resolution or factoring with identical premises and identical consequence may occur twice on the same path in any derivation. In addition, deletion rules, splitting, and the deduction rules are applied in this order, except that splitting is not applied to clauses which contain a selected literal.

All clauses occurring in the clausal form of $\mathrm{Def}_{\Lambda_m(\varphi)}(\pi_r(\varphi, x))$ for a modal formula in $\mathbf{K}_n$ have one of the forms described in Figure 4 [49, 119]. The literals marked with $^+$ are selected in the clauses by the maximal selection function S. The notation $\mathcal{P}(s)$ in the figure represents some literal with a unary predicate symbol and argument term s, and $\mathcal{R}(s,t)$ represents some literal with a binary predicate symbol and argument terms s and t (not necessarily in this order). Two occurrences of $\mathcal{P}(s)$ or $\mathcal{R}(s,t)$ need not be identical, for example, $\neg Q_\psi(x) \vee P_p(x) \vee Q_\chi(x)$ is an instance of $\neg Q_\psi(x) \vee \mathcal{P}(x) \vee \mathcal{P}(x)$, while $\neg Q_\psi(x) \vee \neg R_i(y,x) \vee Q_\chi(y)$ is an instance of $\neg Q_\psi(x) \vee \neg \mathcal{R}(x,y) \vee \mathcal{P}(y)$.

As all non-unit clauses of a typical input set contain a selected literal, all definitional clauses

$\mathcal{P}(a)$	
$\neg Q_\psi(x)^+ \vee \neg P_p(x)^+$	if $\psi = \neg p$
$\neg Q_\psi(x)^+ \vee \mathcal{P}(x)\ [\vee \mathcal{P}(x)]$	if $\psi = \phi_1 \wedge \phi_2$ $[\psi = \phi_1 \vee \phi_2]$
$\neg Q_\psi(x)^+ \vee \neg \mathcal{R}(x,y)^+\ [\vee \mathcal{P}(y)]$	if $\psi = [i]\bot$ $[\psi = [i]\phi]$
$\neg Q_\psi(x)^+\ [\vee \mathcal{P}(f(x))]$	if $\psi = \langle i \rangle \top$ $[\psi = \langle i \rangle \phi]$
$\neg Q_\psi(x)^+ \vee \mathcal{R}(x, f(x))$	

Figure 4. Schematic clausal forms for $\mathbf{K}_n$

can only be used as negative premises of resolution steps. To begin with, there is only one candidate for a positive premise, namely, the ground unit clause $Q_\varphi(a)$ (which representing the input formula φ). Inferences with such ground unary unit clauses produce ground clauses consisting of positive literals only, which are split into ground unit clauses. It can be shown that maximally split (non-empty) inferred clauses have one of two forms: $\mathcal{P}(s)$, or $\mathcal{R}(s, f(s))$, where s is a ground term [119]. In general, s is a nested non-constant functional ground term, which is typically avoided in resolution decision procedures based on an ordering refinement because, in most situations, nesting causes unbounded computations. For the class of clauses under consideration, however, any derived clause is smaller than its positive parent clauses with respect to a well-founded ordering which reflects the structure of the formula.

THEOREM 6 ([119, 121]). *Let φ be a $\mathbf{K}_n$ formula and let N be the clausal form of the formula* $\mathrm{Def}_{\Lambda_m(\varphi)}(\pi_r(\varphi, x))$. *Then*

1. *φ is unsatisfiable in $\mathbf{K}_n$ iff there is a refutation of N by R^{hyp}, and*
2. *any R^{hyp} derivation from N terminates.*

THEOREM 7 ([49]). *Let φ be a $\mathbf{K}_n$ formula. The space complexity for testing the satisfiability of a modal formula φ with R^{hyp} is bounded by $O(nd^m)$, where n is the number of symbols in φ, d is the number of different diamond subformulae in φ, and m is the modal depth of φ.*[2]

Formulae in $\mathbf{K}_n$ translate by the relational translation into the guarded fragment, in particular, into the two-variable guarded fragment $\mathbf{GF}^2$. It is not difficult to see that formulae in $\mathbf{K}_n$ are in fact translated into the subfragment $\mathbf{GF}^-$, introduced in [134]. Under the assumption that either (i) there is a bound on the arity of predicate symbols in $\mathbf{GF}^-$ formulae, or (ii) that each subformula of a $\mathbf{GF}^-$ formula has a bounded number of free variables, the satisfiability problem of $\mathbf{GF}^-$ is PSpace-complete, the same as for the satisfiability problem of $\mathbf{K}_n$. Obviously, there is a bound of two on the arity on predicate symbols occurring in the relational translation of modal formulae in $\mathbf{K}_n$. From these observations a well-known result follows.

THEOREM 8. *The computational complexity of the satisfiability problem of $\mathbf{K}_n$ is PSpace-complete.*

In [81] it is shown that R^{hyp} can be implemented as a modification of the main procedure of a standard (saturation based) first-order theorem prover with splitting (e.g. (M)SPASS [120, 174, 192, 194]) to provide a space optimal decision procedure for $\mathbf{GF}^-$. A direct consequence is the following.

THEOREM 9 ([81, 180]). *R^{hyp} can be turned into a polynomial space resolution decision procedure for $\mathbf{K}_n$.*

A more detailed description of how this can be done is given in Section 3.4.

Another interesting aspect of R^{hyp} is that it can polynomially simulate tableau algorithms [118, 119, 121]. In general, a proof system $\mathcal{A}$ *polynomially simulates* (*p-simulates*) a proof system $\mathcal{B}$ iff there is a function g, computable in polynomial time, mapping proofs of any given formula φ in $\mathcal{B}$ to proofs of φ in $\mathcal{A}$ [38]. To establish a correspondence between tableau proofs and derivations in R^{hyp}, we make use of the fact that each subformula ψ of a given modal formula φ corresponds to a predicate symbol Q_ψ in $\mathrm{Def}_{\Lambda_m(\varphi)}(\pi_r(\varphi, x))$. Every node w occurring in a tableau completion tree corresponds to a term t_w occurring in a R^{hyp} derivation. A formula ψ

[2]The *modal depth* of a formula φ is the maximal nesting of modal operators $\langle i \rangle$ or $[i]$ in φ.

occurring in a set labelling a node w corresponds to a unit clause $Q_\psi(t_w)$ and any edge between nodes w and v with label i in a completion tree corresponds to a unit clause $R_i(t_w, f(t_w))$ in a R^{hyp} derivation, where t_w is the term corresponding to node w and $f(t_w)$ is the term corresponding to node v, for some function symbol f. Given these correspondences, each application of a tableau expansion rule to a completion tree can be simulated by at most two applications of expansion rules in a R^{hyp} derivation.

This p-simulation result extends to tableau algorithms for many extension of $\mathbf{K}_n$, for example extensions by the modal axiom schemas $\mathbf{T}$, $\mathbf{D}$, $\mathbf{B}$, $\mathbf{4}$, and $\mathbf{5}$ [121]. It also extends to other forms of tableau and sequent-style calculi.

The notion of p-simulation leaves open the possibility that an algorithm based on the proof system $\mathcal{A}$ which p-simulates a proof system $\mathcal{B}$ would have to search a much larger search space to find a proof for a given formula than an algorithm based on $\mathcal{B}$. For R^{hyp}, however, it is possible to show that the search space corresponds to that of the tableau algorithm for $\mathbf{K}_n$ presented in Section 4.2 [121]. Related simulation results of tableau procedures for description logics can be found in [118, 119], see also [65]. All these simulation results provide valuable insights into the similarities and difference between tableau methods and resolution. On the one hand, the view presented is that many tableau algorithms are essentially hyperresolution with lazy translation to first-order logic. On the other hand, because of the generality of the setting (first-order logic) it is even possible to exploit the close link with hyperresolution and use it as a basis for systematically developing new tableau procedures. Using this approach, a new tableau decision procedure was essentially ‘read off’ in [49] from a translation-based hyperresolution decision procedure for an expressive PDL-style modal logic.

For the modal logic $\mathbf{K}_n$, an improved version of the relational translation is presented in [9]. In the original presentation, this translation consists of two steps, first mapping a formula from one multi-modal logic into another, and then applying the relational translation to it. Our presentation merges both steps into one. We uniquely associate a unary predicate symbol $P_{p,\sigma}$ with every propositional variable p and sequence σ of modalities. Similarly, we uniquely associate a binary predicate symbol R_σ with every sequence σ of modalities. Then the *tree(-based) relational translation* π_{tr} is defined as follows.

$$
\begin{aligned}
\pi_{tr}(\top, x, \sigma) &= \top & \pi_{tr}(\bot, x, \sigma) &= \bot \\
\pi_{tr}(p, x, \sigma) &= P_{p,\sigma}(x) & \pi_{tr}(\neg\varphi, x, \sigma) &= \neg\pi_{tr}(\varphi, x, \sigma) \\
\pi_{tr}(\varphi \star \psi, x, \sigma) &= \pi_{tr}(\varphi, x, \sigma) \star \pi_{tr}(\psi, x, \sigma) \quad \text{for } \star \in \{\wedge, \vee, \rightarrow, \leftrightarrow\} \\
\pi_{tr}([i]\varphi, x, \sigma) &= \forall y(R_{\sigma.i}(x, y) \rightarrow \pi_{tr}(\varphi, y, \sigma.i)) \\
\pi_{tr}(\langle i\rangle\varphi, x, \sigma) &= \exists y(R_{\sigma.i}(x, y) \wedge \pi_{tr}(\varphi, y, \sigma.i))
\end{aligned}
$$

The translation of a modal formula is given by $\pi_{tr}(\varphi, x, \epsilon)$. The tree relational translation can be viewed as incorporating a feature of the (optimised) functional translation into the relational translation. Whereas the relational translation uses a family of binary predicate symbols R_i, where i is a modal parameter, the tree relational translation uses a larger family of binary predicate symbols R_σ, where σ is a sequence of modal parameters representing a path from the initial world, to encode transitions between worlds. Another difference is that in the tree-based translation the σ are also encoded into the unary predicates.

THEOREM 10. *A modal formula φ is satisfiable in $\mathbf{K}_n$ iff $\pi_{tr}(\varphi, x, \epsilon)$ is first-order satisfiable.*

If we restrict ourselves to $\mathbf{K}$, that is, our logic has only one modality, then the sequence σ in the definition of π_{tr} only serves as a unary coding of the natural numbers. Thus, we can further

simplify the translation by using $\pi_{tr}(\varphi, x, 0)$ as translation of a modal formula and modifying the translation π_{tr} as follows.

$$\pi_{tr}([i]\varphi, x, \sigma) = \forall y(R_{\sigma+1}(x, y) \rightarrow \pi_{tr}(\varphi, y, \sigma{+}1))$$
$$\pi_{tr}(\langle i\rangle\varphi, x, \sigma) = \exists y(R_{\sigma+1}(x, y) \wedge \pi_{tr}(\varphi, y, \sigma{+}1))$$

All other cases in the definition of π_{tr} remain unchanged. In [8] it is shown that we can use the following ordering to ensure that derivations in $R_S^{\succ}$ from the clausal form of $\pi_{tr}(\varphi, x, 0)$ of a $\mathbf{K}$ formula φ terminates: $P_\sigma(s_1, \ldots, s_n) \succ Q_\delta(t_1, \ldots, t_m)$ if either $\sigma < \delta$, or $\sigma = \delta$ and $n > m$. This result can easily be extended to $\mathbf{K}_n$ by defining the ordering $\succ$ as $P_\sigma(s_1, \ldots, s_n) \succ Q_\delta(t_1, \ldots, t_m)$ if either $\mathit{length}(\sigma) < \mathit{length}(\delta)$ or $\mathit{length}(\sigma) = \mathit{length}(\delta)$ and $n > m$. This ordering restriction can be seen to force a kind of top-down approach.

THEOREM 11. *Let φ be a modal formula in $\mathbf{K}_n$ and let N be the clausal form of $\pi_{tr}(\varphi, x, \epsilon)$. Then any derivation from N in R^{ord} (up to redundancy) without splitting terminates.*

One of the interesting aspects of this result is that it does not require the use of structural transformation (nor does it require the use of the splitting rule, but condensing is crucial).

3.2 *Global satisfiability, non-logical axioms, transitive modalities, and* $\mathbf{K4}_n$

So far we have focused on *local satisfiability*, that is, the problem whether for a given modal formula φ, there exists a model $\mathfrak{M} = \langle W, R, V\rangle$ and a world $w \in W$ such that $\mathfrak{M}, w \models \varphi$. Now we turn to the problem of determining whether there is a model $\mathfrak{M}$ such that for all worlds $w \in W$ $\mathfrak{M}, w \models \varphi$, i.e. is φ globally true in some model. The modifications necessary to allows us to determine the *global satisfiability* of a modal formula in $\mathbf{K}_n$ based on the relational translation are minimal: φ is globally satisfiable in $\mathbf{K}_n$ iff $\forall x\, \pi_r(\varphi, x)$ is first-order satisfiable. Is it straightforward to see that the clausal form N of $\mathrm{Def}_{\Lambda_m(\varphi)}(\forall x\, \pi_r(\varphi, x))$ still consists only of $\mathbf{DL}^*$ clauses. Consequently, R^{ord} can decide the satisfiability of the clause set N.

THEOREM 12. *Let Σ be an arbitrary set of axiom schemas such that $\mathbf{K}_n\Sigma$ is complete and the clausal form of the relational frame properties F_Σ corresponding to the axiom schemas in Σ is in $\mathbf{DL}^*$. Let φ be a modal formula in $\mathbf{K}_n$ and let N be the clausal form of $F_{\Sigma,\varphi} = F_\Sigma \wedge \mathrm{Def}_{\Lambda_m(\varphi)}(\forall x\, \pi_r(\varphi, x))$. Then*

1. *φ is not globally satisfiable in $\mathbf{K}_n\Sigma$ iff $F_{\Sigma,\varphi}$ is not first-order satisfiable iff there is a refutation of N by R^{ord},*
2. *N is a set of $\mathbf{DL}^*$ clauses, and*
3. *any derivation from N in R^{ord} (up to redundancy) terminates in double exponential time and in exponential time, if Σ is empty.*

For local and global $\mathbf{K}_n$-satisfiability w.r.t. to a background theory of non-logical axioms the same result is true. Furthermore the complexity of ordered resolution is optimal.

THEOREM 13. *Let $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ be a finite set of of modal formulae and let φ be a modal formula. Let $F_{\Gamma,\varphi}$ be the first-order formula $\exists x\, \pi_r(\varphi, x) \wedge \bigwedge_{i=1,\ldots,n} \forall x\, \pi_r(\gamma_i, x)$ and let N be the clausal form of $\mathrm{Def}_{\Lambda_{\Gamma,\varphi}}(F_{\Gamma,\varphi})$, where $\Lambda_{\Gamma,\varphi}$ contains all non-atomic positions of $F_{\Gamma,\varphi}$. Then*

1. *φ is unsatisfiable in $\mathbf{K}_n$ w.r.t. Γ iff $F_{\Gamma,\varphi}$ is first-order unsatisfiable iff there is a refutation of N by R^{ord},*

2. *N is a set of $\mathbf{DL}^*$ clauses, and*

3. *any derivation from N in $\mathcal{R}^{\mathrm{ord}}$ (up to redundancy) terminates in exponential time.*

In contrast to $\mathcal{R}^{\mathrm{ord}}$, derivations in $\mathcal{R}^{\mathrm{hyp}}$ from the clausal form of $\mathrm{Def}_{\Lambda_{\Gamma,\varphi}}(F_{\Gamma,\varphi})$, as defined in Theorem 13, are not guaranteed to terminate. Tableau algorithms face the same problem, and the solution typically used is a technique called *blocking*. See Sections 4.3 and 4.4 for details. This technique can be transferred to the context of first-order clausal logic and $\mathcal{R}^{\mathrm{hyp}}$ derivations as described in [118]. It involves the addition of a *blocking rule* which at certain points during a derivation adds equations $t_1 \approx t_2$ between ground terms t_1 and t_2 to the clause set, rendering inferences on literals involving the greater of the two terms redundant. One of the interesting properties of this approach is that completeness follows immediately from the general completeness result for $\mathcal{R}^{\mathrm{hyp}}$ [18], only soundness needs to be established. Another way of combining blocking with $\mathcal{R}^{\mathrm{hyp}}$ is presented in [27]. In addition, optimisations techniques like *lazy unfolding* and *absorption*, which will be discussed in detail in Section 4.4, are in-built and therefore free in $\mathcal{R}^{\mathrm{hyp}}$.

However, for $\mathbf{K}_n$ extended with axiom schemas sometimes quite different approaches are required. For example the formula $\forall xyz\,((R_i(x,y) \wedge R_i(y,z)) \rightarrow R_i(x,z))$ stating the transitivity of R_i is not a formula in any of the relevant decidable first-order fragments. The corresponding clause does not belong to $\mathbf{DL}^*$ either. To handle transitive modal logics one possibility is to use the *ordered chaining calculus* introduced in [15] for binary relations satisfying the general schema $R_i \circ R_j \subseteq R_k$. A decision procedure for a first-order fragment covering the modal logics $\mathbf{K4}$, $\mathbf{KD4}$, and $\mathbf{KT4}$, and their multi-modal variants, which is based on ordered chaining, is presented in [77]. Recent work in [124] presents an extension of $\mathcal{R}^{\succ}_{S}$ which can decide the guarded fragment with transitive guards. This provides a decision procedure for all modal logics translatable into this fragment.

In the following we present another approach, the *axiomatic translation* approach [181], which allows a variety of modal logics with transitive modalities to be embedded in $\mathbf{DL}^*$. Consequently this allows the use of $\mathcal{R}^{\mathrm{ord}}$ to decide these logics. This method is not restricted to transitive modal logics and applies to a large class of modal logics.

Remember that structural transformation introduces for each modal subformula $[i]\psi$ of a modal formula φ a predicate symbol $Q_{[i]\varphi}$ in $\pi_r(\varphi, x)$. The general principle of the axiomatic translation approach for $\mathbf{K4}_n$ is the following. For every transitive modality $[i]$ and every subformula $[i]\psi$ of the formula φ, add the first-order formula

$$(A_{\mathbf{4}}) \qquad \forall xy\,((Q_{[i]\psi}(x) \wedge R_i(x,y)) \rightarrow Q_{[i]\psi}(y)).$$

to the translation. The main technical question with the axiomatic translation principle is to know how many instances of such a ‘schema formula’ need to be added to the translation. In the Hilbert axiomatisation, axioms such as $\mathbf{4}$ are valid for all substitution instances. Since we do not have access to a substitution rule, we need to make sure from the outset that enough instances of the schema formulae are present in the translation of φ. (Of course, this does not preclude a lazy implementation which delays the translation of subformulae and the inclusion of instances of schema formulae until absolutely necessary.) The clausal form of $A_{\mathbf{4}}$ is $\neg Q_{[i]\psi}(x) \vee \neg R_i(x,y) \vee Q_{[i]\psi}(y)$, which is a $\mathbf{DL}^*$ clause (and a guarded clause).

THEOREM 14. *Let φ be a modal formula and Ξ the set of all subformulae of the form $[i]\psi$ of φ. Let $F_{\mathbf{4}}$ be the first-order formula $\bigwedge_{[i]\psi\in\Xi} \forall xy\,((Q_{[i]\psi}(x) \wedge R_i(x,y)) \rightarrow Q_{[i]\psi}(y))$. Let N be the clausal form of $F_{\mathbf{4},\varphi} = F_{\mathbf{4}} \wedge \mathrm{Def}_{\Lambda_m(\varphi)}(\pi_r(\varphi, x))$. Then*

1. *φ is unsatisfiable in $\mathbf{K4}_n$ iff $F_{\mathbf{4},\varphi}$ is first-order unsatisfiable iff there is a refutation of N by R^{ord},*

2. *N is a set of $\mathbf{DL}^*$ clauses, and*

3. *any derivation from N in R^{ord} (up to redundancy) terminates in exponential time.*

The same result is true for global satisfiability in $\mathbf{K4}_n$ and also reasoning with respect to non-logical axioms. Theorem 14 reduces reasoning in $\mathbf{K4}_n$ to reasoning in $\mathbf{K}_n$ with background theories. Consequently, R^{hyp} combined with a blocking rule provides an alternative decision procedure for $\mathbf{K4}_n$.

3.3 *Converse modalities and the modal logics $\mathbf{KB}_n$ and $\mathbf{KB4}_n$*

Extending the results of Section 3.1 to modal logics with *converse modalities* or to the modal logics $\mathbf{KB}_n$ and $\mathbf{KB4}_n$ is straightforward. For converse modalities we have to extend our definition of the relational translation π_r as follows:

$$\pi_r([i^\smile]\varphi, x) = \forall y\,(R_i(y,x) \to \pi_r(\varphi, y)) \qquad \pi_r(\langle i^\smile\rangle\varphi, x) = \exists y\,(R_i(y,x) \wedge \pi_r(\varphi, y))$$

Then, Theorem 13 extends to the following.

THEOREM 15. *Let $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ be a finite set of of $\mathbf{K}_n^\smile$ formulae and let φ be a $\mathbf{K}_n^\smile$ formula. Let $F_{\Gamma,\varphi}$ be the first-order formula $\exists x\,\pi_r(\varphi, x) \wedge \bigwedge_{i=1,\ldots,n} \forall x\,\pi_r(\gamma_i, x))$. Let N be the clausal form of $\mathrm{Def}_{\Lambda_{\Gamma,\varphi}}(F_{\Gamma,\varphi})$ where $\Lambda_{\Gamma,\varphi}$ contain all non-atomic positions of $F_{\Gamma,\varphi}$. Then*

1. *φ is unsatisfiable in $\mathbf{K}_n^\smile$ w.r.t. Γ iff $F_{\Gamma,\varphi}$ is first-order unsatisfiable iff there is a refutation of N by R^{ord},*

2. *N is a set of $\mathbf{DL}^*$ clauses, and*

3. *any derivation from N in R^{ord} (up to redundancy) terminates in exponential time.*

In the case of the modal logic $\mathbf{KB}_n$ we extend the relational translation (or the axiomatic translation) by adding the relational frame property $R_{\mathbf{B}}$ corresponding to $\mathbf{B}$, namely $\forall xy\,(R_i(x,y) \to R_i(y,x))$, to the translation of φ. Finally, in the case of $\mathbf{KB4}_n$ we restrict ourselves to the axiomatic translation and again add the relational frame property $R_{\mathbf{B}}$ to the translation of φ.

In all these cases, the clausal form N of the translated modal formulae as well as that of $R_{\mathbf{B}}$ consists only of $\mathbf{DL}^*$ clauses. Consequently, R^{ord} provides us with a decision procedure for the satisfiability of N.

THEOREM 16. *Let φ be a $\mathbf{KB}_n$ formula and let N be the clausal form of the first-order formula $F_{\mathbf{B},\varphi} = \forall xy\,(R_i(x,y) \to R_i(y,x)) \wedge \mathrm{Def}_{\Lambda_m(\varphi)}(\pi_r(\varphi, x))$. Then*

1. *φ is unsatisfiable in $\mathbf{KB}_n$ iff $F_{\mathbf{B},\varphi}$ is first-order unsatisfiable iff there is a refutation of N by R^{ord},*

2. *N is a set of $\mathbf{DL}^*$ clauses, and*

3. *any derivation from N in R^{ord} (up to redundancy) terminates in exponential time.*

The result extends easily to global satisfiability and non-logical axioms. So, the axiomatic translation for $\mathbf{KB}$ is another reduction into $\mathbf{DL}^*$, but also $\mathbf{GF}^2$, and R^{ord} is an exponential time decision procedure [181]. Besides R^{ord} we can also use R^{hyp} to decide the satisfiability of N (this is a consequence of the main results in [82, 83]).

THEOREM 17. *Let φ be a $\mathbf{KB4}_n$ formula and let Ξ be the set of all subformulae of the form $[i]\psi$ of φ. Let $F_{\mathbf{B4}}$ be the first-order formula*

$$\forall xy\,(R_i(x,y) \rightarrow R_i(y,x)) \wedge \bigwedge_{[i]\psi\in\Xi} \forall xy\,((Q_{[i]\psi}(x) \wedge R_i(x,y)) \rightarrow Q_{[i]\psi}(y)).$$

Let N be the clausal form of $F_{\mathbf{B4},\varphi} = F_{\mathbf{B4}} \wedge \mathrm{Def}_{\Lambda_m(\varphi)}(\pi_r(\varphi,x))$. Then

1. *φ is unsatisfiable in $\mathbf{KB4}_n$ iff $F_{\mathbf{B4},\varphi}$ is first-order unsatisfiable iff there is a refutation of N by R^{ord},*
2. *N is a set of $\mathbf{DL}^*$ clauses, and*
3. *any derivation from N in R^{ord} (up to redundancy) terminates in exponential time.*

As described in Section 3.1, in the case of the optimised functional translation, we add so-called *functional frame properties* in the form of (conditional) equations between path terms to accommodate additional axiom schemas like $\mathbf{4}$ and $\mathbf{5}$. Alternatively, one can replace syntactic unification in the inference rules of R with *theory unification* [175, 176]. The resulting calculus is called *theory resolution*. So far, the only decision procedures for modal logics like $\mathbf{K4}_n$ or $\mathbf{KB}_n$ based on theory resolution make use of a term depth bound, that is, any derived clause involving terms of depth greater than a pre-computed bound dependent on the modal formula whose satisfiability is tested will be removed [175, 178].

This section is an incomplete discussion of the different uses of first-order resolution. Due to space restrictions we have only been able to present a few of the translations that are available and have omitted a lot of details. Other translation methods are surveyed in [64, 148]. See also the surveys [65, 49, 122, 180, 182].

3.4 Implementation and optimisation

In this section, we give a brief overview of the implementation of the resolution calculus presented in Section 3.1 and discuss some of the issues involved in using such an implementation for theorem proving in modal logic. For further details on the implementation of first-order theorem provers see e.g. [193, 169, 183].

The procedure ResolutionProver presented in Figure 5 is the main procedure implementing the calculus $R^{\succ}_{S}$. The input is a set N of clauses. The output on termination is a proof of unsatisfiability or a saturated clause set. The procedure operates on two sets of clauses, $\mathcal{US}$ and $\mathcal{WO}$ (the set of *usable clauses* and the set of *worked-off clauses*). The set $\mathcal{WO}$ contains all the clauses that have already been used as premises in inference steps (or can never be used as premises) and the set $\mathcal{US}$ contains all the clauses that still need to be considered as premises. In our particular case, the input set N is the clausal form of the translation of some modal formula.

The procedure proceeds as follows. First, the input set N is simplified by the function ired, that is, all tautologies and strictly subsumed clauses are deleted from N (this is achieved by the two argument ired function). The set N is then divided into two sets: the *usable clauses* $\mathcal{US}$ and the *worked-off clauses* $\mathcal{WO}$. The set $\mathcal{US}$ contains all the clauses which are candidates for

```
Procedure ResolutionProver(N)
local WO, US, NEW, Given;
begin
  WO := ∅;
  US := ired(N,N);
  Stack := emptystack();
  while (US ≠ ∅) and (⊥ ∉ US or not stackempty(Stack))
  do
    if (⊥ ∈ US) then
      (Stack, US, WO) := backtrack(Stack, US,WO);
    else
      begin
        (Given, US) := choose(US);
        if (splittable (Given)) then
          begin
            NEW := firstsplitcase(Given);
            Stack := push(Stack, secondsplitcase(Given))
          end
        else
          begin
            WO := WO ∪ {Given};
            NEW := inf(Given, WO);
          end
      end
      (NEW, WO, US) := ired(NEW, WO, US);
  return(US)
end
```

Figure 5. Standard inference loop in a saturation theorem prover

inferences and the set $\mathcal{WO}$ contains all the clauses that have already been selected for inferences. Initially, the set $\mathcal{WO}$ is the empty set, while $\mathcal{US}$ contains all clauses of N remaining after application of ired. Next the procedure enters the main inference loop in which it remains while the set $\mathcal{US}$ is not empty and the empty clause $\perp$ has not been derived or there are still alternative branches of the derivation tree that need to be considered. Within the main loop it is first checked whether the set $\mathcal{US}$ contains the empty clause. If so, the current branch of the derivation is a closed branch and backtracking takes the computation to a different branch of the derivation. Otherwise the function choose selects a clause from $\mathcal{US}$. This clause is called the *given clause*. If the splitting rule can be applied to the given clause, one of its two split components is taken to be the newly derived clause, which is stored in $\mathcal{NEW}$, and the other split component is pushed onto a stack. Basically, this creates a new branch in the derivation tree that is explored later, if it turns out that the current branch can be closed. This corresponds to a depth-first construction of the derivation tree. If the splitting rule cannot be applied, we add the given clause to the set $\mathcal{WO}$ and compute all conclusions of inferences by resolution and factoring between the given clause and clauses in $\mathcal{WO}$ using the function inf. After removing redundant clauses from the sets $\mathcal{US}$, $\mathcal{WO}$, as well as the newly derived clauses (this is achieved by the three argument ired function), the remaining new clauses are added to the set $\mathcal{US}$, and a new iteration of the main

loop is entered.

Important points to note about the ResolutionProver procedure are the following. First, in the main inference loop, the function inf computes all conclusions derivable from the given clause and clauses in $\mathcal{WO}$. For example, suppose we use the R^{hyp} instance of $R^{\succ}_{S}$. Let the set $\mathcal{US}$ contain the clauses $Q_{\langle 1\rangle\varphi}(t)$, $D_0 = \neg Q_{\langle 1\rangle\varphi}(x) \vee R_1(x, f(x))$, and $D_1 = \neg Q_{\langle 1\rangle\varphi}(x) \vee Q_{\varphi}(f(x))$, as well as n clauses of the form $C_{i+1} = Q_{[1]\psi_i}(t)$ and $D_{i+1} = \neg Q_{[1]\psi_i}(x) \vee \neg R_i(x, y) \vee Q_{\psi_i}(y)$, for $1 \leq i \leq n$. If we first choose each D_i, $0 \leq i \leq n{+}1$ they are simply moved to $\mathcal{WO}$, without any new clause being inferred from them. The same is true if we continue by choosing each C_{i+1} in turn. Finally, when we choose $Q_{\langle 1\rangle\varphi}(t)$, the clauses $R_i(t, f(t))$ and $Q_{\varphi}(f(t))$ is computed by inf and moved to $\mathcal{US}$. When $R_1(t, f(t))$ becomes the given clause, inf computes in one step the clauses $Q_{\psi_i}(f(t))$, for $1 \leq i \leq n$. This corresponds to the application of the tableau inference rule

$$\frac{w : \langle i\rangle\varphi \qquad w : [i]\psi_1 \quad \cdots \quad w : [i]\psi_n}{v : \varphi \qquad v : \psi_1 \quad \cdots \quad v : \psi_n}$$

where v is an i-successor of w. However, if we choose clauses starting with D_0, followed by D_1, and then $Q_{\langle 1\rangle\varphi}(t)$, inf infers $R_1(t, f(t))$ and $Q_{\varphi}(f(t))$, corresponding to an application of the $\Diamond$-rule in the tableau algorithm defined in Figure 8. If we proceed by choosing $R_1(t, f(t))$, then each C_{i+1} directly followed by D_{i+1}, inf infers $Q_{\psi_i}(f(t))$, corresponding to a series of applications of the $\Box$-rule in Figure 8. This shows that the way in which clauses are selected by choose gives us added flexibility in how the search for a refutation is directed.

Second, the ordering $\succ$ and the selection function S only influence the function inf without changing what has just been said. Concerning the selection function S the user is able to select among a fixed set of pre-defined selection functions. The selection function which selects every negative literal in any clause is usually included in that set. Concerning the ordering $\succ$, state-of-the-art first-order theorem provers standardly contain implementations of recursive path orderings, lexicographic path orderings or Knuth Bendix orderings, which are parameterised by an ordering on the signature of the input clause set N, which the user can specify. Refinements of the particular ordering $\succ$ defined in Section 3.1 can be obtained by either recursive path orderings or Knuth Bendix orderings (definitions of orderings and ordering extensions can be found in [52]).

Third, the remaining functions in ResolutionProver are firstsplitcase and secondslitcase which basically determine the order in which branches of a derivation tree are investigated. Again, it is possible to exercise control on this order by using some heuristic.

Fourth, the implementation of backtrack has significant influence on the performance of the prover. On the stack we only store the second split component that may need to be considered at a later point, but not the current state of $\mathcal{WO}$ and $\mathcal{US}$. The information required to return $\mathcal{WO}$ and $\mathcal{US}$ to the correct state on backtracking is stored in each clause, allowing us to remove clauses which are no longer derivable and restoring clauses which are no longer redundant after backtracking. When we derive a contradiction it is not necessary to backtrack to the state associated with the split component currently on top of the stack. Instead more intelligent forms of backtracking are possible. For example, the theorem prover SPASS [194] implements *branch condensing*. Here, on backtracking, all first components not used to derive a contradiction are removed from the set $\mathcal{US}$ as well as all the corresponding second split components on the stack. The prover then backtracks to the second split component which is now on top of the stack, removing clauses which are no longer derivable and restoring clauses which are no longer redundant. For further details see [193]. This form of intelligent backtracking is closely related but not identical to *conflict-directed backjumping* [78, 164]. See also Section 4.2.

An alternative to explicit splitting is *splitting through new propositional variables* [168] implemented in the theorem prover VAMPIRE [169] or the generalisation called *separation* in [179]. In the splitting through propositional variables approach, a clause $C \vee D$ with variable-disjoint components C and D is replaced with two clauses $C \vee p$ and $D \vee \neg p$, where p is a new propositional variable called a *split propositional variable*. The ordering $\succ$ and the selection function S are extended to ensure that p is minimal in $D \vee \neg p$ and $\neg p$ is selected in D. This makes it impossible for the clause $C \vee D$ to be derived from the two new clauses and also blocks $D \vee \neg p$ for inferences until we derive a clause in which p is maximal. The derivation of a contradiction from the split component C in explicit splitting then corresponds to the derivation of a clause $E \vee p$ where E consists solely of split propositional variables. If p is maximal in $E \vee p$ we can derive $E \vee D$ which corresponds to backtracking to the branch of the derivation in which D is true. Note that this again is a form of intelligent backtracking since $E \vee p$ is a representation of all the split components involved in deriving a 'contradiction'. Thus, in backtracking we ignore all other splits not represented by a split propositional variable in $E \vee p$. Unlike branch condensing and (conflict-driven) backjumping, however, those splits are still present. A disadvantage of splitting through new propositional symbols is that subsumption and reductions such as unit propagation are not as effective as for explicit splitting. De Nivelle [47] has suggested modifications of the standard inference and redundancy elimination rules which take account of split propositional variables.

Both explicit splitting and splitting through new propositional variables split a clause $C \vee D$ into split components C and D. The two branches of the derivation do not necessarily investigate disjoint sets of Kripke/first-order models. For variants of splitting we have the option to add the negation of C, $\neg C$, to the branch on which D is true. However, in contrast to propositional logic, the benefit is less obvious. For example, assume that C is $Q_{[i]p}(a)$ and that the clause set to which we add $\neg C = \neg Q_{[i]p}(a)$ contains already the unit clause $Q_{[i](p\wedge q)}(a)$. We can propagate the unit clause $\neg C$ to all clauses in the clause set which removes all occurrences of $Q_{[i]p}(a)$ from those clauses. However, the contradiction between $\neg Q_{[i]p}(a)$ and $Q_{[i](p\wedge q)}(a)$ is not detected. This is true even if the clause set contains the definitional clauses $Q_{[i]p}(x) \vee R(x, f(x))$, $Q_{[i]p}(x) \vee \neg P_p(f(x))$, which we can use to derive $R(a, f(a))$ and $\neg P_p(f(a))$. Only when $R(a, f(a))$ and $Q_{[i](p\wedge q)}(a)$ together with the definition clauses $\neg Q_{[i](p\wedge q)}(x) \vee \neg R(x, y) \vee P_p(y)$ (and $\neg Q_{[i](p\wedge q)}(x) \vee \neg R(x, y) \vee P_q(y)$) are used to derive $P_p(f(a))$, is a contradiction detected. Note also that the clause $R(a, f(a))$ might trigger the derivation of a large number of additional clauses which would not be derived in the absence of $\neg Q_{[i]p}(a)$ or its definitional clauses. In general, the computational effort expended to this point might be great without a guarantee that there is a payoff. Termination is however not compromised.

Used as a procedure to test the satisfiability of a $\mathbf{K}_n$ formula φ with any refinement of $R_S^{\succ}$ and any of the translations presented in this section, ResolutionProver requires exponential space in the size of φ. In [81] we have shown how ResolutionProver can be turned into a space optimal decision procedure for the class $\mathbf{GF}^{\,-}$. This modified procedure provides also a polynomial space decision procedure for the relational translation of $\mathbf{K}_n$ and $\mathbf{KB}_n$ formulae. If we focus on $\mathbf{K}_n$, then a simple modification of ResolutionProver as described in Figure 6 is sufficient. The procedure uses an additional local variable t which stores the term we currently focus on. Initially it is the only ground term in N, and is returned by groundTerm. The procedure choose selects the given clauses in a particular order. It starts by choosing non-ground clauses. This transfers all definitional clauses from $\mathcal{US}$ to $\mathcal{WO}$ without any inference steps being performed. Then it selects ground clauses in an order which ensures that the derivation corresponds to a depth-first exploration of the completion tree in a tableau derivation. Finally, ired is modified so

Procedure ResolutionProver(N)
local $\mathcal{WO}, \mathcal{US}, \mathcal{NEW}$, Given, t;
begin
  t := groundTerm(N);
  ...
  (Given, $\mathcal{US}$, t) := choose($\mathcal{US}$, t);
  ...
  ($\mathcal{NEW}, \mathcal{WO}, \mathcal{US}$) :=
      ired($\mathcal{NEW}, \mathcal{WO}, \mathcal{US}$, t);
  ...
end

Procedure choose($\mathcal{US}$, t)
begin
  if ($C \in \mathcal{US}$ where C is non-ground) **then**
    return($C, \mathcal{US} - \{C\}, t$)
  else if ($Q(t) \vee C \in \mathcal{US}$) **then**
    return($Q(t) \vee C, \mathcal{US} - \{Q(t) \vee C\}, t$)
  else if ($R_i(t, s) \in \mathcal{US}$) **then**
    return($R_i(t, s), \mathcal{US} - \{R_i(t, s)\}, s$)
  else if ($R_i(u, v) \in \mathcal{US}$ with v having greatest depth in $\mathcal{US}$) **then**
    return($R_i(u, v), \mathcal{US} - \{R_i(u, v)\}, v$)
end

Figure 6. Modified procedures for a polynomial space decision procedure for $\mathbf{K}_n$

that it removes from $\mathcal{WO}$ all clauses containing argument terms which are not subterms of the term t. This modification ensures that the information on terms which have been fully explored and does not contribute to a refutation is removed, bringing the space requirements down to polynomial space.

3.5 *Other extensions (counting, nominals)*

The modal logics $\mathbf{K}_n^c$ and $\mathbf{K}_n^{\smile,c}$ with graded modalities and the modal logic $\mathbf{K}_n^o$ with nominals can be translated to first-order logic using a number of different embeddings. The simplest one is an extension of the relational translation as follows (the symbol o_i denotes a nominal).

$$\pi_r(\langle i\rangle_m\varphi, x) = \exists y_1 \ldots y_m\, (R_i(x, y_1) \wedge \ldots \wedge R_i(x, y_m) \wedge y_1 \not\approx y_2 \wedge \ldots \wedge y_{m-1} \not\approx y_m)$$
$$\pi_r([i]_m\varphi, x) = \forall y_1 \ldots y_{m+1}\, ((R_i(x, y_1) \wedge \ldots \wedge R_i(x,y_{m+1})) \rightarrow (y_1 \approx y_2 \vee \ldots \vee y_n \approx y_{m+1}))$$
$$\pi_r(o_i, x) = (x \approx o_i)$$

The superposition calculus [14] and the basic superposition calculus [17] are extensions of $R_S^{\succ}$ with rules for equality reasoning. In [113, 112] it is shown that the basic superposition calculus can be used to decide the satisfiability of knowledge bases in the $\mathcal{SHIQ}$ description logic. It follows that it can also be used to decide the satisfiability of formulae in $\mathbf{K}_n^c$ and $\mathbf{K}_n^{\smile,c}$.

An extension of the optimised functional translation to $\mathbf{K}_n^c$ is presented and shown to be sound and complete in [150].

4 TABLEAU-BASED ALGORITHMS

In this section, we describe tableau-based decision procedures for modal logics and discuss their complexity and implementation issues. First, we discuss various choices for presenting tableau algorithms in general, and then present the basic tableau algorithm for $\mathbf{K}_n$ together with a detailed discussion of implementation and optimisation issues. Next, we modify this algorithm to handle $\mathbf{K4}_n$, background theories, converse modalities, and their combinations, and point out relevant modifications concerning the implementation and optimisation.

Intuitively, a tableau algorithm tries to construct, for an input formula φ, a model of φ; i.e., to decide the validity of a formula ψ, the tableau algorithm is started with $\neg\psi$. Depending on the modal logic, it is often convenient to consider an abstraction of models rather than models, namely a so-called *tableau*.

4.1 Tableau algorithms in general

We start with a description of a tableau algorithm for multi modal $\mathbf{K}_n$. Roughly speaking, this algorithm takes the input formula φ and deduces constraints on the model it is going to build by breaking it down into its sub-formulae. We will first describe different styles in which this attempt at a model construction has been described and relate them to each other.

The "breaking down" is realized through *tableau expansion rules*; quite often, we find one such expansion rule per logical constructor. For example, if we know that $\psi_1 \wedge \psi_2$ should be true in world w of the model we are constructing, then we break the conjunction down and explicitly add the constraints that each ψ_i has to be true in w. Next, we discuss the rules that handle box- and diamond formulae. Intuitively, if we know that $\langle i\rangle\psi$ should be true in world w, then we "generate" a witness world w', which is i-accessible from w and in which ψ is true. This can be formalised in different ways:

- for certain modal logics such as $\mathbf{K}_n$, one can first handle all formulae talking about a single world, then collect *all* constraints concerning another world and process these, and so on [129]. This approach is sometimes called "distructive" [92] (see also Chapter 2) because we can forget the constraints concerning an "old" world once we procede to the next one.

- *labelled* tableaux are closely related to propositional tableaux: they are sets of *labelled* formulae that partially describe a model: each formula is labelled with the world it should be true in (see Chapter 2 of this handbook). For example, the case where $\langle 1\rangle\psi$ is true in world w would translate to finding the labelled formula $w : \langle 1\rangle\psi$ in our tableau, and the $\Diamond$-rule adds labelled formulae $w' : \psi$ and $w1w'$, where the latter encodes that w' is 1-accessible from w. This information is required if we find, additionally, a formula of the form $w : [1]\psi'$: in this case, the $\Box$-rule adds $w' : \psi'$.

  Alternatively, one can store the information that w' is i-accessible from w in the labels by using appropriate sequences instead of atomic "names" w, w'. We start with the empty sequence labelling the input concept, and then append these labels, e.g., as follows: if a world is generated for a labelled formula $s : \langle i\rangle\psi$, we name this world $s(i,\psi)$ and simply introduce the new labelled formula $s(i,\psi) : \psi$. Please note that $si : \psi$ does not suffice because $\langle 1\rangle\psi \wedge \langle 1\rangle\neg\psi$ is satisfiable, but only in a world that has two *distinguished* 1-acccessible worlds.

- other tableau algorithms explicitly store the relational structure of the model (or tableau) they are building. More precisely, they work on labelled graphs (often trees) where nodes represent worlds and labelled edges represent i-accessibility. Moreover, nodes are labelled with the *set of formulae* that should be true in the corresponding world. Thus, instead of finding two labelled formulae $w' : \psi'$ and $w' : \psi''$ in a tableau, we would find both formulae in the label of the node w', written $\{\psi', \psi''\} \subseteq \mathcal{L}(w')$.

  An advantage of this approach is that all information concerning a single world is kept in the same place. For example, it allows for the detection of obvious inconsistencies such as $w : p$ and $w : \neg p$ by a test that is local to $\mathcal{L}(w)$. When considering logics with converse

or graded modalities, the advantages of this "one node per world" approach become even more pronounced. State-of-the-art implementations of modal tableau algorithms adopt this approach [159, 90], which is why we have chosen it for this section.

Similarly, the $\vee$-rule is often formulated using either branching or non-determinism in the model construction. For example, if we know that $\psi_1 \vee \psi_2$ should be true in w, then the $\vee$-rule can be formalised in the following two ways:

- we *branch* our model construction into two, one in which ψ_1 is true in w and one in which ψ_2 is true in w, and then continue with the construction of each branch independently.

  This is how non-deterministic constructors are handled in standard first order and modal logic tableau: the tableau rules expand a tree where each branching represents a non-deterministic choice, and thus where each path stands for a possible model.

- we non-deterministically choose one ψ_i to be true in w. This yields a non-deterministic algorithm which, when implemented, requires back-tracking to be complete.

  From a computational perspective, this approach is preferable since, in contrast to the above "branching" alternative, it preserves the useful "one node per world" property. Additionally, it can easily be adapted to exploit techniques developed for solving SAT problems, such as David-Putnam and related heuristics [41, 87]. State-of-the-art implementations of modal tableau algorithms handle disjunctions (and possibly other non-deterministic operators) in this way, and are combined with intelligent back-tracking (or back-jumping) and heuristics to make the "good" choice first, see Section 4.2.

The algorithms described in this chapter will use the latter, non-deterministic formulation, and will work on a single model/tableau at any world in time, where all information concerning each world is stored in a single node.

Figure 7 shows two example applications of different tableau algorithms to decide the satisfiability of the $\mathbf{K}$ formula $\psi = \langle 1 \rangle p \wedge \langle 1 \rangle q \wedge [1](\neg p \vee \neg q)$. On the left hand side, we show the result of a standard labelled tableau, where we use sequences as labels. First, we have broken down the conjunctions, then generated two new labels $(1, p)$ and $(1, q)$ for the two diamond formulae. Next, we have expanded the box formula for both new worlds, and finally branched for the disjunctions. The resulting tree stands for four different attempts to construct a model, one for each path from a leaf node to the root. Only the one ending in the filled node corresponds to a model since all other branches contain obvious inconsistencies: e.g., the first one contains both $(1, p) : p$ and $(1, p) : \neg p$.

On the right hand side, we show a (successfull) application of the non-deterministic version of a tableau algorithm working on trees. It has generated three nodes w_i with labels that are completely expanded sets of formulae. Here, the edges stand for the accessibility relations, i.e., w_2 and w_3 are 1-accessible from w_1. In contrast, on the left hand side, (the formulae along) one *path* in the tree represents a model, i.e., edges relate formulae that are true in the same model.

4.2 *Local satisfiability for multi modal* $\mathbf{K}_n$

Before we describe the algorithm, we introduce an appropriate data structure in which to represent models (and later tableaux). Firstly, it will be convenient to assume that all formulae descriptions are in *negation normal form* (see Section 2). The tableau algorithms presented in this section work on *completion trees*: a *completion tree* is a finite tree where each node x is

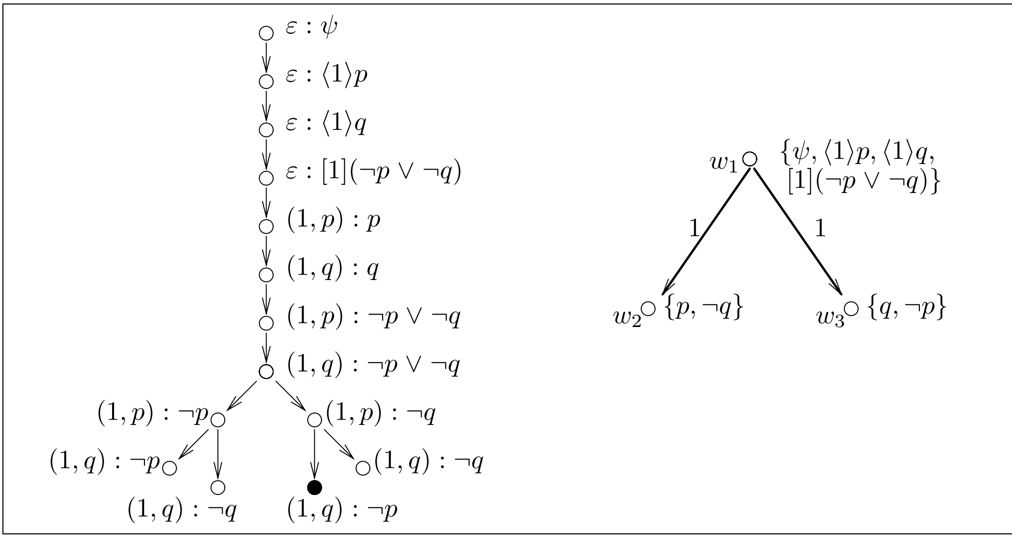


Figure 7. Two application of tableau algorithms to the same formula.

labelled with a *set* $\mathcal{L}(x)$ of formulae, and edges are labelled with modal parameters. A node y is called an *i-successor of* a node x if y is a successor of x and the edge from x to y is labelled with i. A completion tree is said to be *closed* if it contains a node x with $\{p, \neg p\} \subseteq \mathcal{L}(x)$; a completion tree that is not closed is *open*, and it is *complete* if no *expansion rule* applies—the expansion rules are given in Figure 8. Please note that they are formulated in such a way that, if a rule is applicable (i.e., the corresponding condition is satisfied by the current completion tree), then its application indeed changes the tree.

To decide the satisfiability of ϕ (in NNF), the tableau algorithm is started with a completion tree consisting of the root node x_0 only, with $\mathcal{L}(x_0) = \{\phi\}$. It applies the expansion rules until the completion tree becomes closed or complete, and returns "ϕ is satisfiable" if the expansion rules can be applied such that they yield a complete and open tableau, and "ϕ is unsatisfiable" otherwise. The "*can* be applied" formulation is due to the non-deterministic $\vee$-rule, as discussed in Section 4.1. Also, the algorithm does not fix any order in which the rules are to be applied, which means that an implementation has to/can chose a "good" one.

$\wedge$-rule:	*If*	there is a node x with $\psi_1 \wedge \psi_2 \in \mathcal{L}(x)$ and $\{\psi_1, \psi_2\} \not\subseteq \mathcal{L}(x)$,
	then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{\psi_1, \psi_2\}$.
$\vee$-rule:	*If*	there is a node x with $\psi_1 \vee \psi_2 \in \mathcal{L}(x)$ and $\{\psi_1, \psi_2\} \cap \mathcal{L}(x) = \emptyset$,
	then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{\psi_i\}$ for some $i \in \{1, 2\}$.
$\Diamond$-rule:	*If*	there is a node x with $\langle i \rangle \psi \in \mathcal{L}(x)$ and x has no i-successor y with $\psi \in \mathcal{L}(y)$,
	then	create a new i-successor y of x with $\mathcal{L}(y) := \{\psi\}$.
$\Box$-rule:	*If*	there is a node x with $[i]\psi \in \mathcal{L}(x)$ and x has an i-successor y with $\psi \notin \mathcal{L}(y)$,
	then	$\mathcal{L}(y) := \mathcal{L}(y) \cup \{\psi\}$.

Figure 8. The expansion rules for $\mathbf{K}_n$.

Before discussing the properties of this algorithm, we would like to point out that the tableau

rule

$$\frac{w : \langle i \rangle \varphi \qquad w : [i]\psi_1 \quad \cdots \quad w : [i]\psi_n}{v : \varphi \qquad v : \psi_1 \quad \cdots \quad v : \psi_n}$$

mentioned in Section 3.4 of this chapter corresponds, in our notation, to

If there is a node x with $\{\langle i \rangle \varphi, [i]\psi_1, \ldots, [i]\psi_n\} \subseteq \mathcal{L}(x)$,
then create a new i-successor y of x with $\mathcal{L}(y) := \{\varphi, \psi_1, \ldots, \psi_n\}$.

The fact that our algorithm decides satisfiability of $\mathbf{K}_n$ formulae is an immediate consequence of the following lemma, for which we first need to define the semantics of completion trees. Let $\mathbf{T}$ be a completion tree, $\mathfrak{M} = \langle W, R, V \rangle$ a model, and π a (total) mapping from the nodes of $\mathbf{T}$ to W. Then $\mathfrak{M}$ is said to *satisfy* $\mathbf{T}$ *via* π if, for each node x in $\mathbf{T}$,

1. for each $\psi \in \mathcal{L}(x)$, we have $\mathfrak{M}, \pi(x) \models \psi$ and
2. for each i successor y of x, we have $R_i(\pi(x), \pi(y))$.

LEMMA 18. *Let ϕ be a $\mathbf{K}_n$ formula and $\mathbf{T}$ a completion tree generated by the tableau algorithm for ϕ.*

1 *When applied to ϕ, the tableau algorithm terminates.*

2 *If $\mathfrak{M}$ satisfies $\mathbf{T}$ via π and one of the expansion rules is applicable to $\mathbf{T}$, then this rule can be applied in such a way that it yields a $\mathbf{T}'$ satisfied by $\mathfrak{M}$ via π or an extension of π.*

3 *If $\mathbf{T}$ is complete, then there exists a model $\mathfrak{M}$ and a mapping π such that $\mathfrak{M}$ satisfies $\mathbf{T}$ via π iff $\mathbf{T}$ is open.*

Lemma 18.1 is due to the fact that (i) the breadth and depth of the completion tree are bounded linearly by the length of ϕ, (ii) node labels are sets of subformulae of ϕ, and (iii) the completion tree is built in a monotonic way, i.e., each rule strictly increases node labels or adds new nodes. Property (i) is due to the fact that there are at most $|\phi|$ diamond modalities in ϕ and that the maximal modal depth of formulae in node labels strictly decreases from a node to its (i-)successors. Lemma 18.2 is an immediate consequence of the semantics of $\mathbf{K}_n$ and completion trees. For example, let the $\Diamond$-rule be applicable to some $\mathbf{T}$ with $\langle i \rangle \psi \in \mathcal{L}(x)$, and let $\mathfrak{M}$ satisfy $\mathbf{T}$ via π. Hence $\mathfrak{M}, \pi(x) \models \langle i \rangle \psi$, and thus there exists some $w \in W$ with $R_i(\pi(x), w)$ and $\mathfrak{M}, w \models \psi$. As a consequence, we can extend π to $\pi(y) = w$ for y the newly introduced node, and $\mathfrak{M}$ satisfies the result of this rule application via (the extended) π. The “if” direction of Lemma 18.3 is easy since each open completion tree can be viewed as a model with W the set of nodes, $x \in V(p)$ if $p \in \mathcal{L}(x)$, and $R_i(x, y)$ if y is an i-successor of x. The only-if direction of Lemma 18.3 is trivial.

THEOREM 19. *The $\mathbf{K}_n$ tableau algorithm decides $\mathbf{K}_n$ satisfiability and can be implemented in polynomial space.*

As an immediate consequence of Lemma 18 and the fact that each model $\mathfrak{M}$ satisfying ψ is one that satisfies the initial completion tree (and vice versa), we thus have the first point of Theorem 19. The second part follows from the following observations. As a consequence of (i) and (ii) in the proof sketch of Lemma 18.1, we can store each branch of a completion tree in space bounded polynomially in the length of ψ. Next, we observe that we can consider each branch independently, and thus we can build the completion tree in a depth first manner, keeping only a single branch in memory at each point in time. Finally, our $\vee$-rule is non-deterministic, but it is known how to transform a non-deterministic polynomial space algorithm into a deterministic one that also runs in polynomial space [172].

Implementation Issues

Even this “simplest” modal logic $\mathbf{K}_n$ extends propositional logic, and thus the complexity is rather discouraging from an implementational perspective: we may have to consider a number of models (or completion trees) that is exponential in the size of the input formula. Moreover, because the completion tree is usually built in a depth first manner, with the $\wedge$- and $\vee$-rules being exhaustively applied to a given node before creating any modal successors with the $\Diamond$-rule, it is easy to find formulae with unsatisfiability “hidden” in the leaves of the tree for which a naive implementation will always exhibit pathological worst case behaviour. Consider, for example, the formula:

$$\phi = (p_1 \vee q_1) \wedge \ldots \wedge (p_n \vee q_n) \wedge \langle i \rangle \psi \wedge [i]\neg\psi. \tag{11}$$

There are 2^n different ways in which the combination of the $\wedge$-and $\vee$-rules can be applied to a node whose label is initialised with $\{\phi\}$, but in each case subsequent applications of the $\Diamond$- and the $\Box$-rules will eventually lead to a closed completion tree. A naive implementation of the trace technique with “chronological” backtracking search would consider all 2^n possible expansions before concluding that the input formula is unsatisfiable; this kind of unproductive backtracking search in often referred to as *thrashing*.

Fortunately, a wide range of optimisation techniques has been developed in order to improve the efficiency with which the algorithm explores the space of possible models [104, 103]. Although these optimisations may lead to a situation in which the worst case behaviour would actually be much worse than the theoretical worst case, empirical studies have shown that such optimised algorithms are very effective with *typical* formulae, i.e., formulae derived from applications. These techniques include normalisation and simplification, dependency directed backtracking, SAT based search techniques, simplification of node labels, heuristics and caching.

Normalisation and Simplification As usual, our description of the $\mathbf{K}_n$ tableau algorithm assumes that the input formula is in negation normal form (NNF); this simplifies the (description of the) algorithm, but it means that a completion tree will only be closed when a propositional variable and its negation occur in the same node label. For example, when testing the satisfiability of the formula $(p \wedge q) \wedge \neg(p \wedge q)$, the transformation into NNF would give $(p \wedge q) \wedge (\neg p \vee \neg q)$; in practise this means that, in spite of the “obvious” contradiction, backtracking search will be performed in order to determine that the formula is unsatisfiable.

For this reason, practical algorithms do not transform the input concept into NNF, but include a $\neg$-rule that performs a single (negation) normalisation step (e.g., applying the $\neg$-rule to $\neg(p \wedge q) \in \mathcal{L}(x)$ would cause $\neg p \vee \neg q$ to be added to $\mathcal{L}(x)$), and a completion tree is closed if it contains a node x with $\{\psi, \neg\psi\} \subseteq \mathcal{L}(x)$ for an arbitrary formula ψ. Moreover, in order to facilitate the detection of such closure conditions, the input formula is *normalised* and *simplified* so that logically equivalent formulae are more often syntactically equivalent. This is achieved by (recursively) applying a set of rewrite rules to the input formula, and by ordering conjuncts w.r.t. some total ordering. For example, we re-write $\vee$ and $\Diamond$ formulae as negated $\wedge$ and $\Box$ formulae, respectively; we remove redundant parentheses between conjunctions; we order conjuncts; and we simplify formulae using the following equivalences: $(\psi \wedge \psi) \leftrightarrow \psi$, $\neg\neg\psi \leftrightarrow \psi$, $(\psi \wedge \neg\psi \wedge \rho) \leftrightarrow \neg\top$, $(\psi \wedge \top) \leftrightarrow \psi$, $(\psi \wedge \neg\top) \leftrightarrow \neg\top$, and $[i]\top \leftrightarrow \top$.

If the above transformations are applied to the formula ϕ from the above example (11), then $\langle i \rangle\psi$ would be rewritten as $\neg[i]\neg\psi$, $\neg[i]\neg\psi \wedge [i]\neg\psi$ would be rewritten as $\neg\top$ and $(p_1 \vee q_1) \wedge \ldots \wedge (p_n \vee q_n) \wedge \neg\top$ would be rewritten as $\neg\top$, a formula that is trivially unsatisfiable.

If ψ_1 was added to $\mathcal{L}(x)$ by the		
$\wedge$-rule for $\psi_1 \wedge \psi_2 \in \mathcal{L}(x)$,	then $\mathsf{dep}(\psi_j, x) := \mathsf{dep}(\psi_1 \wedge \psi_2, x)$ for each $j \in \{1, 2\}$	
$\vee$-rule for $\psi_1 \vee \psi_2 \in \mathcal{L}(x)$,	then $\mathsf{dep}(\psi_j, x) := \mathsf{dep}(\psi_1 \vee \psi_2, x) \cup \{b\}$ for each $j \in \{1, 2\}$	
$\Diamond$-rule for $\langle i \rangle \psi_1 \in \mathcal{L}(x')$,	then $\mathsf{dep}(\psi_1, x) := \mathsf{dep}(\langle i \rangle \psi_1, x')$	
$\Box$-rule for $[i]\psi_1 \in \mathcal{L}(x')$,	then $\mathsf{dep}(\psi_1, x) := \mathsf{dep}([i]\psi_1, x') \cup \mathsf{dep}(\langle i \rangle \psi_2, x')$	
	where x was generated by the $\Diamond$-rule for $\langle i \rangle \psi_2 \in \mathcal{L}(x')$	

Figure 9. Inductive definition of $\mathsf{dep}(\psi, x)$

Dependency Directed Backtracking As we saw in the above example (11), inherent unsatisfiability concealed in sub-formulae can lead to large amounts of unproductive backtracking search known as thrashing. Although the normalisation and simplification technique described above solved the problem for this example, this might not have been the case if the unsatisfiability caused by the modal sub-formulae had been slightly less trivial. Consider, e.g., the following, only slightly modified formula ϕ':

$$\phi' = (p_1 \vee q_1) \wedge \ldots \wedge (p_n \vee q_n) \wedge \langle i \rangle (\psi \wedge \rho) \wedge [i]\neg\psi. \tag{12}$$

To avoid an exponential search in the case of ϕ', a more sophisticated solution is required, and can be found by adapting a form of dependency directed backtracking called *backjumping*, which has also been used, e.g., in solving constraint satisfiability problems [19] and (in a slightly different form) in the HARP theorem prover [153].

Intuitively, backjumping works by labelling each formula ψ in the label of a node x with a dependency set $\mathsf{dep}(\psi, x)$ indicating the *branching points* (i.e., applications of the $\vee$-rule) on which it depends. In case the completion tree is closed because it contains some node x with $\{\psi, \neg\psi\} \in \mathcal{L}(x)$, we use $\mathsf{dep}(\psi, x)$ and $\mathsf{dep}(\neg\psi, x)$ to identify the most recent branching point b on which ψ or $\neg\psi$ depends. The algorithm can then *jump* back to b over intervening branching points *without* exploring any alternative branches (non-deterministic choices), and make a different non-deterministic choice which might not lead to the same closure condition being encountered. In case no such b exists, the closure did not depend on any non-deterministic choice, and the algorithm stops.

To be more precise, a *branching point* is simply a non-negative integer b indicating the b-th $\vee$-rule application in the run of the tableau algorithm. Initially, for x_0 the root node and ϕ the input formula, $\mathsf{dep}(\phi, x_0) := \emptyset$. The sets $\mathsf{dep}(\psi, x)$ are then defined inductively as shown in Figure 9. In this way, each formula in each node label is associated with a dependency set. If the completion tree is closed because it contains some node x with $\{\psi, \neg\psi\} \in \mathcal{L}(x)$, the *closure dependency set* $S := \mathsf{dep}(\psi, x) \cup \mathsf{dep}(\neg\psi, x)$, and the algorithm backtracks to the b-th $\vee$-rule application (or exits if $b = 0$).

The procedure for expanding a completion tree $\mathbf{T}$ is given in Figure 10. For an input formula ϕ, $\mathbf{T}$ is initialised to contain a single node x_0 with $\mathcal{L}(x_0) = \{\phi\}$ and $\mathsf{dep}(\phi, x_0) := \emptyset$; ϕ is satisfiable if Satisfiable($\mathbf{T}$,0) returns $\{-1\}$ and unsatisfiable otherwise. For example, when expanding the formula ϕ' from 12 above, the $\wedge$-rule might first be applied exhaustively via recursive calls to Satisfiable, resulting in $\{p_1 \vee q_1, \ldots, p_n \vee q_n, \langle i \rangle (\psi \wedge \rho), [i]\neg\psi\} \subseteq \mathcal{L}(x_0)$ and $\mathsf{dep}(\psi_j, x_0) = \emptyset$ for each formula $\psi_j \in \mathcal{L}(x_0)$. These dependencies reflect the fact that, so far, no non-deterministic choices have been made. A top-down and "left to right" strategy might then cause Branch to be called n times, with, for the j-th call, $b = j$, $f_1 = p_j$, $f_2 = q_j$ and

```
Procedure Satisfiable(T,b)
local f;
begin
  if for some node x in T, {ψ, ¬ψ} ∈ L(x)
  then
    return(dep(ψ, x) ∪ dep(¬ψ, x))
  else if T is complete then
    return({−1})
  else
    begin
      f := some unexpanded formula in node
      x in T
      if f is of the form ψ1 ∨ ψ2 then
        return(Branch(T,b                    +
        1,x,ψ1,ψ2,dep(f, x)))
      else
        begin
          expand f (as per Fig. 8 and 9)
          return(Satisfiable(T,b))
        end
    end
end
```

```
Procedure Branch(T,b,x,f1,f2,D)
local S, T-saved;
begin
  T-saved := T
  add f1 to L(x) with dep(f1, x) = {b}∪D
  S := Satisfiable(T,b)
  if b ∉ S then
    return(S)
  else
    begin
      T := T-saved
      add f2 to L(x) with dep(f2, x) =
      b ∪ d
      return(S ∪ Satisfiable(T,b))
    end
end
```

Figure 10. Procedure for tableau expansion with backjumping

$D = \emptyset$, so that $p_1, \ldots, p_n$ are added to $\mathcal{L}(x_0)$ with $\mathsf{dep}(p_j, x_0) = j$. Next, recursive calls to Satisfiable would expand: $\langle i\rangle(\psi \wedge \rho) \in \mathcal{L}(x_0)$, causing the generation of an i-successor x_1 of x_0 with $\mathcal{L}(x_1) = \{\psi \wedge \rho\}$ and $\mathsf{dep}(\psi \wedge \rho, x_1) = \emptyset$; $[i]\neg\psi \in \mathcal{L}(x_0)$, causing $\neg\psi$ to be added to $\mathcal{L}(x_1)$, with $\mathsf{dep}(\neg\psi, x_1) = \emptyset$; and $\psi \wedge \rho \in \mathcal{L}(x_1)$, causing ψ and ρ to be added to $\mathcal{L}(x_1)$, with $\mathsf{dep}(\psi, x_1) = \mathsf{dep}(\rho, x_1) = \emptyset$. The completion tree would then be closed, as $\{\psi, \neg\psi\} \subseteq \mathcal{L}(x_1)$, and Satisfiable would return $\mathsf{dep}(\psi, x_1) \cup \mathsf{dep}(\neg\psi, x_1) = \emptyset$.

If we were using chronological backtracking, the recursion would return to the n-th branching point, i.e., the one where Branch was called with $b = n$, $f_1 = p_n$ and $f_2 = p_n$. $\mathbf{T}$ would be restored to its state prior to adding p_n to $\mathcal{L}(x_0)$, and the rule would be applied again such that q_n was added to $\mathcal{L}(x_0)$. Using backjumping, however, we return from Branch immediately because $b \notin S$. This is obviously true for all of the preceding branching points, so all calls to Branch will return without expanding the completion trees obtained by adding the various q_j to $\mathcal{L}(x_0)$, and Satisfiable will eventually return $\emptyset$, allowing us to conclude that ϕ' is unsatisfiable.

SAT Based Search Techniques Even with the addition of dependency directed backtracking, a naive implementation of the $\Diamond$-rule is inherently inefficient as it can lead to the repetition of parts of the expansion. For example, given an input formula

$$\phi'' = (\rho \vee \psi_1) \wedge \ldots \wedge (\rho \vee \psi_n), \tag{13}$$

where $\psi_1 \wedge \ldots \wedge \psi_n$ is satisfiable but ρ is not, the procedure described above would lead to the construction of n (possibly large) closed completion trees, each with $\rho \in \mathcal{L}(x_0)$, before a complete and open completion tree is constructed.

This problem can be avoided by using more sophisticated search techniques. One of best

known of these is the Davis-Putnam algorithm, originally designed for solving propositional satisfiability (SAT) problems [42]. The basic idea behind Davis-Putnam is that, instead of branching on unexpanded disjunctions, we branch on a formula ψ such that ψ occurs in an unexpanded disjunction in a node x of the completion tree and $\{\psi, \neg\psi\} \cap \mathcal{L}(x) = \emptyset$; the algorithm then searches the two possible trees obtained by adding ψ or $\neg\psi$ to $\mathcal{L}(x)$. This basic technique is usually enhanced with heuristics and simplification rules (which we will discuss in more detail below); in particular, we usually branch first on formulae that occur in many unexpanded disjunctions and, if $\{\psi \vee \rho, \neg\psi\} \subseteq \mathcal{L}(x)$, then $\psi \vee \rho$ is deterministically expanded by adding ρ to $\mathcal{L}(x)$. It is easy to see that if this strategy is applied to ϕ'' above, we would branch first on ρ (as it occurs in n unexpanded disjunctions), and at most one closed completion tree (if ρ is tried first) would be constructed before finding a complete and open one.

This technique has been shown to be very effective with formulae generated at random using generators adapted from those used to generate SAT problems [87, 114]. Such problems typically include a relatively small number of propositional variables (so there is likely to be significant repetition of the sub-formulae occurring in disjunctions), and have a very low modal depth (so the importance of propositional reasoning is emphasised); this is because large numbers of propositional variables and/or a high modal depth would result in almost all problems of reasonable size being trivially satisfiable. Formulae from applications, however, typically do not exhibit these characteristics, and Davis-Putnam is much less effective—in fact it can even be counter-productive if the negated formulae that Davis-Putnam introduces are large and/or complex [103].

An alternative technique used in [56] is to enhance the standard chronological backtracking method with a *no-good list* for each node, i.e., a set of formulae, each of which has already been shown to lead to a closed completion tree when it is added to the node label by an application of $\vee$-rule. Formulae in the no-good list are not considered when applying the $\vee$-rule. Using this technique with ϕ'' above, ρ would be added to the no-good list after the first application of the $\vee$-rule leads to a closed completion tree. In subsequent applications of the $\vee$-rule, ρ would not be considered, and ψ_j would always be selected. This technique has the advantage that wasted search is avoided without adding negated formulae that could themselves lead to additional (possibly non-deterministic) expansion.

Note that, when using these (and other) optimisations in addition to backjumping, care must be taken to ensure that *all* dependencies are being taken into consideration. For example, when using a no-good list to restrict the possible choices made by the $\vee$-rule, it is important to also consider the dependencies associated with the relevant formulae in the no-good list.

Simplification of Node Labels As well as the standard tableau expansion rules described in Figure 8, additional inference rules can be applied to the formulae occurring in a node label, usually with the objective of simplifying them and reducing the number of $\vee$-rule applications. The most commonly used simplification, often called *Boolean Constraint Propagation* (BCP) [74], is again derived from SAT solvers, where it is usually used in conjunction with the Davis-Putnam procedure. The basic idea is to identify a disjunction $\psi_1 \vee \ldots \vee \psi_n \in \mathcal{L}(x)$ such that the negations of all but one of the ψ_j are already elements of $\mathcal{L}(x)$; when this is the case, the formula can be deterministically expanded by adding the relevant ψ_j to $\mathcal{L}(x)$. This amounts to applying the following inference rule

$$\frac{\neg\psi_1, \ldots, \neg\psi_n, \psi_1 \vee \ldots \vee \psi_n \vee \psi}{\psi}$$

to the formulae in a node label, which is a restricted variant of hyper resolution, see 3.1.

As we have already seen, when $\neg\rho$ is added to $\mathcal{L}(x_0)$ during the expansion of ϕ'' above, the BCP rule can be applied to all the remaining $\phi \vee \psi_j$ formulae, leading to a complete and open completion tree without any further applications of the $\vee$-rule. Note that, as with the more sophisticated search techniques described above, careful consideration needs to be given to the dependencies of formulae added by such inference rules if they are to be used together with backjumping.

Heuristics As mentioned in Section 4.1, one advantage of the non-deterministic formulation of the $\vee$-rule is that an algorithm can try to choose a "good" order in which to try the different possible expansions. In practise, this usually means using heuristics to select the way in which the $\vee$-rule is applied to the disjunctions in a node label, and the order in which the successor nodes created by $\Diamond$-rule applications are expanded; in either case, a heuristic function is used to compute the relative "goodness" of candidate formulae/nodes.

When using the Davis-Putnam technique, the well known MOMS heuristic [74] is often used to select the formulae on which to branch; it tries to select formulae that will maximise the effect of BCP and so minimise the number of non-deterministic choices needed in order to complete the completion tree [103]. There is little evidence, however, that (a suitably adapted form of) this heuristic is effective with modal formulae, and even some evidence to suggest that interference with the backjumping optimisation makes it counter productive [103].

An alternative heuristic, whose design was prompted by this observation, tries to maximise the effect of backjumping by preferentially selecting formulae with low valued dependencies [103, 99]. This heuristic has the added advantage that it can also be used to select the order in which successor nodes are expanded.

Caching When using the top-down construction strategy, all information from predecessors is added to a node label before it is processed. This means that, when a given node has been fully expanded (i.e., the expansion rules have been exhaustively applied to it), a successor node y with $\mathcal{L}(y) = \{\psi_1, \ldots, \psi_n\}$ can be treated as an independent problem, equivalent to testing the satisfiability of $\psi_1 \wedge \ldots \wedge \psi_n$.

A completion tree may contain many such nodes, and the labels of nodes tend to be quite similar, particularly as the labels of i-successors of a node x each contain the same formulae resulting from $\Box$-rule applications to $[i]\psi$-formulae in $\mathcal{L}(x)$. For some formulae, this may result in the same sub-problem being solved again and again. In order to avoid this, it is possible to cache and re-use the results of such sub-problems. The usual technique is to use a hash table to store the satisfiability status of node labels (i.e., sets of formulae treated as a conjunction). Before applying any expansion rules to a new node x, the cache is interrogated to determine if the satisfiability status of $\mathcal{L}(x)$ is already known. If it is known, then the result can be used without further expansion, i.e., $\mathcal{L}(x)$ can be treated as though it were either $\{\bot\}$ (for unsatisfiable) or $\{\top\}$ (for satisfiable). If the satisfiability status of $\mathcal{L}(x)$ is not known, then $\mathcal{L}(x)$ is added to the cache, and its status set to satisfiable if a complete and open completion tree rooted in x can be constructed, and to unsatisfiable otherwise.

Since the satisfiability of a set of formulae L implies the satisfiability of each subset of L, and the unsatisfiability of a set of formulae L implies the unsatisfiability of each superset of L, this basic idea can be extended to check for satisfiable supersets of $\mathcal{L}(x)$ and unsatisfiable subsets of $\mathcal{L}(x)$. However, this requires a considerably more sophisticated data structure if cache operations are to be efficient [100, 86].

Apart from the problem of the storage required for the cache, another more subtle disadvantage of caching is that, in the case where the cache returns "unsatisfiable" for $\mathcal{L}(x)$, there is no information about the cause of the unsatisfiability that can be used to derive the dependency

information required for backjumping. Backjumping can still be performed by combining the dependency sets of all of the formulae in $\mathcal{L}(x)$, but this is likely to overestimate the set of branching points on which the unsatisfiability depends.

Another useful form of caching is a technique known as *model merging* [103, 91]. The idea here is to prove the satisfiability of a node label $\mathcal{L}(x)$ by showing that open and complete completion trees for $L_1, \ldots, L_k$ with $L_1 \cup \ldots \cup L_k = \mathcal{L}(x)$ can be combined into an open and complete completion tree for $\mathcal{L}(x)$ by simply "gluing" their root nodes together. This is possible if there are no "interactions" between the various completion trees, e.g., if there are no j, k such that either $\psi \in L_j$ and $\neg\psi \in L_k$ or $\langle i \rangle \psi \in L_j$ and $[i]\rho \in L_k$ for some i, ψ and ρ. Thus model merging involves (a) cashing satisfiable sets of formulae that occur as root labels of open and complete completion trees, and (b) trying to prove the satisfiability of some $\mathcal{L}(x)$ by finding cached sets L_j that do not interact in the above sense.

4.3 Transitive modalities and $\mathbf{K4}_n$

The main problem one has to overcome when modifying the $\mathbf{K}_n$ tableau algorithm presented in Section 4.2 to $\mathbf{K4}_n$ is *termination*. Please recall that the $\mathbf{K}_n$ tableau algorithm terminates "automatically" since it builds a tree of bounded depth and breadth in a monotonic way. As we will see, this is not the case for $\mathbf{K4}_n$. Consider, e.g., the $\mathbf{K4}_n$ formula $\langle i \rangle \psi \wedge [i]\langle i \rangle \psi$. A $\mathbf{K4}_n$ tableau algorithm would start with a root node x_0 labelled with this formula, then apply the $\wedge$-rule, and then generate an i-successor x_1. Next, the $\Diamond$-rule would be applicable and it would add $\langle i \rangle \psi$ to $\mathcal{L}(x_1)$. Thus the $\Diamond$-rule would generate an i-successor x_2 of x_1. At his point, the difference between $\mathbf{K}_n$ and $\mathbf{K4}_n$ becomes apparent: in $\mathbf{K4}_n$ models, R_i has to be transive, and thus x_2 should be i-accessible from x_0, i.e., $\langle i \rangle \psi$ would also need to be true in (the world represented by) x_2. Hence, we would need a (new) rule that adds $\langle i \rangle \psi$ to $\mathcal{L}(x_2)$. However, this would trigger the applicability of the $\Diamond$-rule, which would generate an i-successor x_3 of x_2. Now we can use R_i's transitivity again to argue that $\langle i \rangle \psi$ needs to be added to $\mathcal{L}(x_3)$, and continue the whole pattern to construct an infinite i-chain. Thus the tableau algorithm would not terminate: in contrast to the $\mathbf{K}_n$ tableau algorithm, the maximal modal depth of formulae in node labels no longer decreases from a node to its successors.

To regain termination, we observe that, creating this infinite path, we keep repeating the same actions. More precisely, the node labels of the nodes $x_1, x_2, \ldots$ are all identical. In the following, we show how we can prevent this "looping" using a cycle detection mechanism called "blocking". Intuitively, after the creation of x_2 and the application of the $\Box$-rule, we could have noticed that $\mathcal{L}(x_2) = \{\psi, \langle i \rangle \psi\} = \mathcal{L}(x_1)$, and decided to *not* apply the $\Diamond$-rule to x_2 because (i) we would continue repeating ourselves and (ii) it is not necessary since $\mathcal{L}(x_1) = \mathcal{L}(x_2)$ implies that we can use (the world represented by) x_1 for the world represented by x_2. The latter means that we can build a model $\mathfrak{M}$ with $(x_1, x_1) \in R_i$ in which $\mathfrak{M}, x_1 \models \psi$ and, from the semantics, $\mathfrak{M}, x_1 \models \langle i \rangle \psi$.

The other problem we have to overcome is how we are going to take care of R_i's transitivity. Consider an i-successor y of a node x. Now if y has in turn an i-successor z, then this situation represents a model in which $(x, z) \in R_i$, i.e., z "should" also be an i-successor of x. That is, if $[i]\psi \in \mathcal{L}(x)$, then ψ should be in $\mathcal{L}(z)$. One possible way to achieve this would be to give up on working on completion *trees*, and instead work on graphs where we would add the additional i-edge between x and z. For implementation purposes, however, trees are clearly advantagous, and thus we choose to employ an alternative technique: the same effect as adding the additional i-edge between x and z can be obtained by adding $[i]\psi$ to $\mathcal{L}(y)$ for each $[i]\psi \in \mathcal{L}(x)$. This will

be realized in a modified $\Box$-rule.

Now we formalise this in our tableau algorithm. First, we define the notion of a *blocked* node. We use ancestors and offsprings in the usual way; a node x is *directly blocked* if it has an ancestor x' with $\mathcal{L}(x) \subseteq \mathcal{L}(x')$; a node is *blocked* if it is directly blocked or if it has an ancestor that is directly blocked. Next, we take this notion into account in the $\mathbf{K4}_n$ expansion rules, which are given in Figure 11. Compared to the $\mathbf{K}_n$ expansion rules, these expansion rules only apply to nodes that are not blocked (however, $\Box$-rule can add formulae to the label of a directly blocked node), and the $\Box-$rule "pushes" box formulae in the way discussed above.

$\wedge$**-rule:**	*If*	there is a node x that is not blocked with $\psi_1 \wedge \psi_2 \in \mathcal{L}(x)$ and $\{\psi_1, \psi_2\} \not\subseteq \mathcal{L}(x)$,
	then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{\psi_1, \psi_2\}$.
$\vee$**-rule:**	*If*	there is a node x that is not blocked with $\psi_1 \vee \psi_2 \in \mathcal{L}(x)$ and $\{\psi_1, \psi_2\} \cap \mathcal{L}(x) = \emptyset$,
	then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{\psi_i\}$ for some $i \in \{1, 2\}$.
$\Diamond$**-rule:**	*If*	there is a node x that is not blocked with $\langle i \rangle \psi \in \mathcal{L}(x)$ and x has no i-successor y with $\psi \in \mathcal{L}(y)$,
	then	create a new i-successor y of x with $\mathcal{L}(y) := \{\psi\}$.
$\Box$**-rule:**	*If*	there is a node x that is not blocked with $[i]\psi \in \mathcal{L}(x)$ and x has an i-successor y with $\psi \notin \mathcal{L}(y)$,
	then	$\mathcal{L}(y) := \mathcal{L}(y) \cup \{\psi, [i]\psi\}$.

Figure 11. The expansion rules for $\mathbf{K4}_n$.

To convince ourselves that this algorithm indeed decides satisfiability of $\mathbf{K4}_n$ formulae, we sketch the same technical lemma as for $\mathbf{K}_n$.

LEMMA 20. *Let ψ be a $\mathbf{K4}_n$ formula and $\mathbf{T}$ a completion tree generated by the tableau algorithm for ψ.*

1. *When applied to ψ, the tableau algorithm terminates.*

2. *If $\mathfrak{M}$ satisfies $\mathbf{T}$ via π and one of the expansion rules is applicable to $\mathbf{T}$, then this rule can be applied in such a way that it yields a $\mathbf{T}'$ satisfied by $\mathfrak{M}$ via (possibly an extension of) π.*

3. *If $\mathbf{T}$ is complete, then there exists a model $\mathfrak{M}$ and a mapping π such that $\mathfrak{M}$ satisfies $\mathbf{T}$ via π iff $\mathbf{T}$ is open.*

Again, we only sketch the proof. Termination is due to the same three observations as in the sketch of Lemma 18, but the reason for the bound of the depth of the tree is more involved (and the bound is now quadratic). Consider three nodes x, y, and z where y is an i-successor of x and z a j-successor of y. If $i \neq j$, then the maximal modal depth of formulae in the label of z is strictly smaller than the one in the label of y. If $i = j$, then either $\mathcal{L}(z) \subseteq \mathcal{L}(y)$ or y was generated for a different different diamond formula in $\mathcal{L}(x)$ than z in $\mathcal{L}(y)$. In the former case, z is blocked. The latter case can only occur linearly often in the length of the input formula. As a consequence, paths in the completion tree are of length at most quadratic in the length of the input formula. Lemma 20 (ii) is similar to the $\mathbf{K}_n$ case, but we have to exploit the transitivity of R_i to explain why pushing $[i]\psi$ from a node to its i-successor preserves $\mathfrak{M}$ being a model via π. Finally, the construction of a model from an open, complete completion tree in Lemma 20 (iii) is slightly modified: firstly, only un-blocked nodes represent worlds in the model. Secondly, we also add (x, y') to R_i if x has an i-successor y which is blocked and y' is an ancestor of y with

$\mathcal{L}(y) \subseteq \mathcal{L}(y')$. Thirdly, we extend R_i so that it is transitively closed, i.e., if $\{(x, y), (y, z)\} \subseteq R_i$, then we also have $(x, z) \in R_i$.

The same reasons as for $\mathbf{K}_n$ then yield the following theorem.

THEOREM 21. *The $\mathbf{K4}_n$ tableau algorithm decides $\mathbf{K4}_n$ satisfiability and can be implemented in polynomial space.*

Implementation Issues

As we have seen, the main difference between the tableau algorithms for $\mathbf{K}_n$ and $\mathbf{K4}_n$ is the introduction of blocking. In fact the blocking condition described above, which specifies a subset relationship between the labels of blocked and blocking nodes, is already optimised w.r.t. the one originally described in [93], which specified label equality. The subset condition means that blocking can occur sooner, thus avoiding possibly costly expansion.

Consider, for example, a node x labelled as follows:

(14) $$\mathcal{L}(x) = \{\rho, \psi, \langle i \rangle \psi, [i]\langle i \rangle \psi\},$$

With subset blocking, an i-successor y of x with $\mathcal{L}(y) = \{\psi, \langle i \rangle \psi, [i]\langle i \rangle \psi\}$ would be blocked by x; with equality blocking, a block would not be established until an i-successor z of y is constructed, with $\mathcal{L}(z) = \mathcal{L}(y)$. This may lead to significant additional work if ψ is itself a large and/or complex formula.

Apart from blocking, the algorithm is very similar to the $\mathbf{K}_n$ case, and most of the optimisation techniques described in Section 4.2 can be applied without modification. Blocking does, however, mean that additional care is required when caching and re-using the satisfiability of a set of formulae, because the satisfiability of the set of formulae in the label of a blocked node is contingent on the satisfiability of the set of formulae in the label of the blocking node [103]. This dependency also extends to the satisfiability of the sets of formulae in the labels of any nodes on the path between the blocking node and the blocked node.

Consider, for example, a node x labelled as in 14 above, where ψ is unsatisfiable. As we have seen, an application of the $\Diamond$-rule to $\langle i \rangle \psi \in \mathcal{L}(x)$, followed by applications of the $\Box$-rule to $[i]\langle i \rangle \psi \in \mathcal{L}(x)$, would lead to the creation of an i-successor y of x with $\mathcal{L}(y) = \{\psi, \langle i \rangle \psi, [i]\langle i \rangle \psi\}$, and no expansion rule would be applicable to y as it would be blocked by x. Updating the cache to indicate that the set of formulae $\mathcal{L}(y)$ is satisfiable would, however, clearly be an error, as ψ is unsatisfiable.

4.4 Non-logical axioms and background theories

Now that we have understood how to handle transitivity in $\mathbf{K4}_n$, understanding how to handle background theories is easy. Consider the satisfiability of a formula ϕ w.r.t. the background theory $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$, and remember that the nodes of our completion tree represent worlds of the model we are trying to build, which has to be a common model of ϕ and Γ. Moreover, $\psi \in \mathcal{L}(x)$ stands for the fact that ψ is true in the world (represented by) x. As before, at least one node (the root node) will carry ϕ in its label. Additionally, we will make sure that all nodes will carry each γ_i in their label. As a consequence, we will have a similar problem with termination as we have seen for $\mathbf{K4}_n$, i.e., the maximal modal depth of formulae in node labels does no longer decrease from a node to its successor. Fortunately, we can use the same blocking technique as for $\mathbf{K4}_n$: a node x is *directly blocked* if it has an ancestor x' with $\mathcal{L}(x) \subseteq \mathcal{L}(x')$, and it is *blocked* if it is directly blocked or if it has an ancestor that is directly blocked. The expansion rules for

$\mathbf{K}_n$ w.r.t. background theories are given in Figure 12: they contain the $\mathbf{K}_n$ $\Box$−rule, an additional Γ-rule that adds Γ to each node label, and the $\mathbf{K4}_n$ restriction to blocked nodes. We call the resulting algorithm the *extended* $\mathbf{K}_n$ tableau algorithm.

$\wedge$**-rule:**	*If*	there is a node x that is not blocked with $\psi_1 \wedge \psi_2 \in \mathcal{L}(x)$ and $\{\psi_1, \psi_2\} \not\subseteq \mathcal{L}(x)$,
	then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{\psi_1, \psi_2\}$.
$\vee$**-rule:**	*If*	there is a node x that is not blocked with $\psi_1 \vee \psi_2 \in \mathcal{L}(x)$ and $\{\psi_1, \psi_2\} \cap \mathcal{L}(x) = \emptyset$,
	then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{\psi_i\}$ for some $i \in \{1, 2\}$.
$\Diamond$**-rule:**	*If*	there is a node x that is not blocked with $\langle i \rangle \psi \in \mathcal{L}(x)$ and x has no i-successor y with $\psi \in \mathcal{L}(y)$,
	then	create a new i-successor y of x with $\mathcal{L}(y) := \{\psi\}$.
$\Box$**-rule:**	*If*	there is a node x that is not blocked with $[i]\psi \in \mathcal{L}(x)$ and x has an i-successor y with $\psi \notin \mathcal{L}(y)$,
	then	$\mathcal{L}(y) := \mathcal{L}(y) \cup \{\psi\}$.
Γ**-rule:**	*If*	there is a node x that is not blocked with $\Gamma \not\subseteq \mathcal{L}(x)$,
	then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \Gamma$.

Figure 12. The expansion rules for $\mathbf{K}_n$ with background theories.

We can state and prove an analogous technical lemma as for $\mathbf{K}_n$ and $\mathbf{K4}_n$, and then use the same reasons to conclude the following theorem.

THEOREM 22. *The extended* $\mathbf{K}_n$ *tableau algorithm decides satisfiability w.r.t background theories.*

However, in contrast, we no longer can implement our tableau algorithm in polynomial space: firstly, it is known that satisfiability of $\mathbf{K}_n$ formulae w.r.t. background theories is ExpTime-complete (we can adapt the proofs in [68, 162]). Secondly, we can easily construct a formula and a background theory such that each of their model contains a path of length exponential in the input formulae: we can use propositional variables $p_1, \ldots, p_\ell$ as a "binary counter" for numbers between 0 and $2^\ell - 1$, and non-logical axioms to enforce that, if the p_i at a world w represent a number k, then the p_i at a world w' with $(w, w') \in R_i$ represent the number $k + 1 \mod \ell$. Thirdly, in the worst case, our algorithm indeed constructs completion trees that are of depth exponential in the length of the input formulae: for $\mathbf{K4}_n$, we could argue that the maximal modal depth decreases from a node to a j-successor of its i-successor (if $i \neq j$). For $\mathbf{K}_n$ with background theories, this is no longer true. As a consequence of this exponential length and the non-deterministic $\vee$-rule, our tableau algorithm runs, in the worst case, in non-deterministic double exponential time—which is clearly sub-optimal. In [56], an optimal tableau algorithm for (the description logic) variant of $\mathbf{K}_n$ with background theories was presented; however, to the best of our knowledge, this algorithm has never been implemented, whereas the sub-optimal one described here has proven to work surprisingly well in practice [159, 90].

Implementation Issues

One obvious consequence of the above algorithm is that expansion of the formulae in Γ occurs in every node in the completion tree, and this can easily lead to an explosion in the size of the completion tree or in the number of different possible completion trees that can be (non-deterministically) constructed for a given input formula. For example, if $\Gamma = \{\langle i \rangle \psi_1, \ldots, \langle i \rangle \psi_n\}$, a completion tree containing $nn! + 1$ nodes will be constructed. Similarly, if $\gamma \in \Gamma$, with

$\gamma = ((p_1 \vee q_1) \wedge \ldots \wedge (p_n \vee q_n))$, and the input formula leads to the construction of a completion tree containing k nodes, then there are 2^{kn} different ways to apply the $\wedge$- and $\vee$-rules to the resulting k copies of γ. This explosion in the size of the search space can easily lead to a catastrophic degradation in performance, even when optimisations such as backjumping and caching are employed [102].

Fortunately, optimisations known as *lazy unfolding* and *absorption* have proved to be very effective in reducing the size of the search space, particularly for background theories derived, e.g., from class based knowledge representation formalisms.

Lazy Unfolding In background theories, formulae are often (restricted to be) of the form $p \rightarrow \psi$ or $p \leftrightarrow \psi$ for some propositional variable p. A theory

$$\Gamma = \{p_1 \leftrightarrow \psi_1, \ldots, p_\ell \leftrightarrow \psi_\ell, p_{\ell+1} \rightarrow \psi_{\ell+1}, \ldots, p_{\ell+m} \rightarrow \psi_{\ell+m}\}$$

is said to be *unfoldable*, if it satisfies the following conditions.

- Formulae in Γ are *unique*. I.e., for each propositional variable p, Γ contains at most one formula of the form $p \leftrightarrow \psi$ (i.e., $p_i \neq p_j$ for $1 \leqslant i < j \leqslant \ell$), and if it contains a formula of the form $p \leftrightarrow \psi$, then it does not contain any formulae of the form $p \rightarrow \psi$. (Note that an arbitrary set of formulae $\{p \rightarrow \psi_1, \ldots, p \rightarrow \psi_n\}$ can be combined into a single formula $p \rightarrow (\psi_1 \wedge \ldots \wedge \psi_n)$.)

- Γ is *acyclic*. I.e., there is no formula $p_i \leftrightarrow \psi_i \in \Gamma$ such that p_i occurs either directly or indirectly in ψ_i.[3] A propositional variable p occurs indirectly in a formula ψ if there is a propositional variable formula p' such that p' occurs directly in ψ, and there is a formula $p' \leftrightarrow \psi' \in \Gamma$ such that p occurs either directly or indirectly in ψ'.

Instead of being dealt with using the Γ-rule, such a set of formulae can be lazily *unfolded* during the tableau expansion. I.e., for a formula $p_1 \rightarrow \psi_1 \in \Gamma$, if p_i is added to $\mathcal{L}(x)$ for some node x, then ψ_i is also added to $\mathcal{L}(x)$, and for a formula $p_j \leftrightarrow \psi_j \in \Gamma$, if p_j ($\neg p_j$) is added to $\mathcal{L}(x)$ for some node x, then ψ_j (resp. $\neg\psi_j$) is also added to $\mathcal{L}(x)$.

It is obvious that an arbitrary background theory Γ can be divided into an unfoldable part Γ_u and a general part Γ_g such that $\Gamma_u \cup \Gamma_g = \Gamma$ and $\Gamma_u \cap \Gamma_g = \emptyset$. The unfoldable part Γ_u can then be dealt with using lazy unfolding while the general part Γ_g is dealt with using the Γ-rule.

In fact it has been shown that the definition of an unfoldable theory can be extended somewhat while still allowing the use of the above lazy unfolding technique. In particular, the formulae occurring on the left hand side of (bi-) implications can also be negated propositional variables, and the acyclicity condition can be relaxed by distinguishing positive and negative occurrences of propositional variables in a stratified theory [109, 132].

Absorption Given the effectiveness of lazy unfolding in dealing with the unfoldable part of a background theory Γ, it makes sense to try to rewrite the formulae in Γ so that the size of Γ_g can be reduced. Absorption is just such a rewriting optimisation.

The idea behind absorption derives from the observation that (apparently non-unfoldable) formulae in Γ_g are often of the form $p \wedge \rho \rightarrow \psi$. This formula can be rewritten as $p \rightarrow (\psi \vee \neg\rho)$, which allows it to be moved from Γ_g to Γ_u, provided that Γ_u does not already contain a formula of the form $p \leftrightarrow \psi'$. In case Γ_u does contain such a formula, then the technique can be extended by using the formulae in Γ_u to perform further rewriting. E.g., if $p \leftrightarrow \psi_i \in \Gamma_u$ and $p \rightarrow \psi_j \in \Gamma_g$, then the second formula can be rewritten as $\psi_i \rightarrow \psi_j$ and, if ψ_i is of the form $q \wedge \psi_i'$, the formula

[3] For the purposes of lazy unfolding, only cycles consisting entirely of $\leftrightarrow$ axioms are problematical.

can be further rewritten as $q \to \psi_j \vee \neg\psi_i'$. A more detailed description of the various re-writings used in absorption can be found in [109].

4.5 Converse modalities

So far, our tableau algorithms only use expansions rules that are either local to a single node, create new successors, or push formulae from a node label into the label of a *successor*. The objective of this section is to discuss a tableau algorithm for $\mathbf{K}_n^{\smile}$, i.e., $\mathbf{K}_n$ with converse modalities. It is well-known that satisfiability in $\mathbf{K}_n^{\smile}$ can be polynomially reduced to the satisfiability of $\mathbf{K}_n$ w.r.t. background theories [43]. However, from an implementation perspective, this approach is not feasible since it leads to a dramatic performance degradation, and we thus present a direct algorithm.

As mentioned in Section 2, $\mathbf{K}_n^{\smile}$ requires reasoning in both ways over relations R_i. For our tableau algorithm, this will simply mean that we push formulae up and down in a completion tree. To realize this, we define the notion of an i-neighbour, which requires a few other concepts: firstly, to avoid numerous case distinction, we introduce a function $\mathsf{Cv}(\cdot)$ on modal parameters as follows:[4] $\mathsf{Cv}(i) = i^{\smile}$ and $\mathsf{Cv}(i^{\smile}) = i$. Next, we consider completion trees where each edge is labelled with a possibly converse modal parameter i or $i^{\smile}$. Finally, for α a (possibly converse) modal parameter, we call a node y an *α-neighbour* of a node x if y is an α-successor of x or if x is a $\mathsf{Cv}(\alpha)$-successor of y. The expansion rules for $\mathbf{K}_n^{\smile}$ are identical to those for $\mathbf{K}_n$, with the only difference being that the $\Box$- and the $\Diamond$-rules now consider α-neighbours instead of α-successors (but the $\Diamond$-rule still generates an α-successor if no appropriate α-neighbour is available); they can be found in Figure 13.

$\wedge$-rule:	*If*	there is a node x with $\psi_1 \wedge \psi_2 \in \mathcal{L}(x)$ and $\{\psi_1, \psi_2\} \not\subseteq \mathcal{L}(x)$,
	then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{\psi_1, \psi_2\}$.
$\vee$-rule:	*If*	there is a node x with $\psi_1 \vee \psi_2 \in \mathcal{L}(x)$ and $\{\psi_1, \psi_2\} \cap \mathcal{L}(x) = \emptyset$,
	then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{\psi_i\}$ for some $i \in \{1, 2\}$.
$\Diamond$-rule:	*If*	there is a node x with $\langle\alpha\rangle\psi \in \mathcal{L}(x)$ and x has no α-neighbour y with $\psi \in \mathcal{L}(y)$,
	then	create a new α-successor y of x with $\mathcal{L}(y) := \{\psi\}$.
$\Box$-rule:	*If*	there is a node x with $[\alpha]\psi \in \mathcal{L}(x)$ and x has an α-neighbour y with $\psi \notin \mathcal{L}(y)$,
	then	$\mathcal{L}(y) := \mathcal{L}(y) \cup \{\psi\}$.

Figure 13. The expansion rules for $\mathbf{K}_n^{\smile}$.

We can state and prove an analogous technical lemma as for $\mathbf{K}_n$, and then use similar reasons to conclude the first part of following theorem.

THEOREM 23. *The $\mathbf{K}_n^{\smile}$ tableau algorithm decides $\mathbf{K}_n$ satisfiability and can be implemented in polynomial space.*

To implement the $\mathbf{K}_n^{\smile}$ tableau algorithm in polynomial space, we can use the following "restart" technique: for each node x, we first apply the $\wedge$- and the $\vee$-rule exhaustively.[5] Next, if a formula is added to $\mathcal{L}(x)$ by the $\Box$-rule for some $[\alpha]\psi$ in a $\mathsf{Cv}(\alpha)$-successor of x, then we disregard the whole sub-tree below x and re-start its construction from scratch. As a consequence of this "strategy", all branches of a completion tree are independent, and we can still construct a completion tree depth first.

[4] Remember that $\mathbf{K}_n$ with converse modalities provides modal parameters i and $i^{\smile}$ for $1 \leq i \leq m$.

[5] Please note that we never made any assumptions or restrictions on the order in which the rules are to be applied.

Implementation Issues

Although the restart technique can be used to enable $\mathbf{K}_n^{\smile}$ completion trees to be constructed using a depth first strategy, the technique is not used in practice as rebuilding discarded parts of the completion tree can be very costly (and space usage is rarely a problem in practice). Without this technique, however, extra care is required when using some of the optimisation techniques described above.

Without the depth first strategy, the satisfiability of (the formula represented by) the label of a node x can no longer be treated as an independent problem, because the results of expanding x might affect its predecessor (unless x is the root node). This means that, although we can re-use cached unsatisfiability results from the cache as before, we must either disregard satisfiable results, or use more sophisticated caching techniques (e.g., storing additional information that would allow us to check for possible interactions with the predecessor node) [103].

Computation of the dependencies used in backjumping is also made more difficult by the loss of the depth first strategy. In particular we need to consider the dependency set of the $\langle i \rangle$ formula in x that led to the generation of an i-successor y in order to compute $\mathsf{dep}(\psi, y)$ when ψ is added to $\mathcal{L}(y)$ as a result of a $\Box$-rule application to a formula $[i]\psi \in \mathcal{L}(x)$. With depth first expansion, this is usually accomplished by combining $\Diamond$-rule applications with all relevant $\Box$-rule applications. Without depth first expansion, this is usually achieved by extending the labelling of either nodes or edges with the dependency set of the $\Diamond$-formula that caused them to be added to the completion tree.

Finally, without the depth first strategy it is necessary, in general, to save the state of the whole completion tree at each $\vee$-rule application (as mentioned above, the depth first strategy allows state saving and restoring to be restricted to a single node label). This problem can be ameliorated by using a lazy state saving strategy, where node labels are only saved when they are about to be extended by some rule application.

4.6 Converse modalities and background theories

In the last sections, we have seen how to extend the basic $\mathbf{K}_n$ tableau algorithm to a decision procedure for $\mathbf{K4}_n$, for $\mathbf{K}_n$ with background theories, and for $\mathbf{K}_n^{\smile}$. For the first two extensions, we discussed a technique to "artificially" ensure termination while preserving soundness and completeness. For the third extension, we introduced the concept of *neighbours* and modified the expansion rules as to work up and down the completion tree. In this section, we will put these techniques and concepts together—and show that their combination requires a further adjustment.

To be more precise, in this section, we discuss a tableau algorithm for $\mathbf{K}_n^{\smile}$ with background theories, i.e., converse modal parameters can occur both in the input formula and in the formulae of the background theory. Next, we discuss the expansion rules, which are given in Figure 14. Clearly, in the presence of converse modal parameters, we use the notion of α-neighbours. Similarly, in the presence of background theories, we use the Γ-rule, and we use blocking to ensure termination. However, the combination of background theories with converse modal parameters requires two modifications. Consider an i-successor y of x with $[i^{\smile}]\psi \in \mathcal{L}(y)$, and assume that y is blocked and x is not blocked. Hence there is some node y' with $\mathcal{L}(y) \subseteq \mathcal{L}(y')$. In case the tableau algorithm stops with an open, complete completion tree, we will try to construct a model $\mathfrak{M}$ from this tree, and we will have $(x, y') \in R_i$. Now $\mathcal{L}(y) \subseteq \mathcal{L}(y')$ implies that $[i^{\smile}]\psi \in \mathcal{L}(y')$, and we thus have to show that $\mathfrak{M}, x \models \psi$. However, if we would not apply the $\Box$-rule to y

because y is blocked, we might not find $\psi \in \mathcal{L}(x)$, and thus our construction might fail. This observations leads to the first modification:

1. we call a node *indirectly blocked* if it is blocked, and if its predecessor is blocked as well. Then we apply all but the $\Diamond$-rule to nodes that are not indirectly blocked.

In our example case, y was indirectly blocked, and thus the $\Box$-rule would add ψ into $\mathcal{L}(x)$. Next, consider some $[i^{\smile}]\psi' \in \mathcal{L}(y') \setminus \mathcal{L}(y)$. The same reasons as for $[i^{\smile}]\psi$ imply that we should find $\psi' \in \mathcal{L}(x)$—which we would not since our blocking condition only requires $\mathcal{L}(y) \subseteq \mathcal{L}(y')$. This observation leads to the second modification:

2. a node x is *directly blocked* if it has an ancestor x' with $\mathcal{L}(x') = \mathcal{L}(x)$.

For obvious reasons, we refer to the former blocking condition as *subset blocking*, and to this new condition as *equality blocking*. Please note that, in this setting, it is unavoidable that blocking is "dynamic", that is, a blocked node can later become not blocked. In contrast, with a certain strategy for the order of rule applications, this can be avoided in the $\mathbf{K}_n$ case.

$\wedge$**-rule:**	*If*	there is a node x that is not indirectly blocked with $\psi_1 \wedge \psi_2 \in \mathcal{L}(x)$ and $\{\psi_1, \psi_2\} \not\subseteq \mathcal{L}(x)$,
	then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{\psi_1, \psi_2\}$.
$\vee$**-rule:**	*If*	there is a node x that is not indirectly blocked with $\psi_1 \vee \psi_2 \in \mathcal{L}(x)$ and $\{\psi_1, \psi_2\} \cap \mathcal{L}(x) = \emptyset$,
	then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{\psi_i\}$ for some $i \in \{1, 2\}$.
$\Diamond$**-rule:**	*If*	there is a node x that is not blocked with $\langle\alpha\rangle\psi \in \mathcal{L}(x)$ and x has no α-neighbour y with $\psi \in \mathcal{L}(y)$,
	then	create a new α-successor y of x with $\mathcal{L}(y) := \{\psi\}$.
$\Box$**-rule:**	*If*	there is a node x that is not indirectly blocked with $[\alpha]\psi \in \mathcal{L}(x)$ and x has an α-neighbour y with $\psi \notin \mathcal{L}(y)$,
	then	$\mathcal{L}(y) := \mathcal{L}(y) \cup \{\psi\}$.
Γ**-rule:**	*If*	there is a node x that is not indirectly blocked with $\Gamma \not\subseteq \mathcal{L}(x)$,
	then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \Gamma$.

Figure 14. The expansion rules for $\mathbf{K}_n^{\smile}$ with background theories.

4.7 *Other extensions (counting, nominals, transitive closure, and fixpoints)*

In this section, we discuss two other extensions of our tableau algorithms. Firstly, we discuss $\mathbf{K}_n^{\smile,o}$, the extension of $\mathbf{K}_n^{\smile}$ with background theories and *nominals*. Secondly, we discuss $\mathbf{K}_n^c$, the extension of $\mathbf{K}_n$ with graded modalities, and also how to ensure termination in the additional presence of background theories. Finally, we discuss modal logics with a transitive closure operator and fixpoints.

Further add nominals to $\mathbf{K}_n^{\smile}$ with background theories

$\mathbf{K}_n^{\smile,o}$ with background theories is of interest because it lacks the tree models property and because it requires another form of non-local reasoning. The former point was already discussed in Section 2. To see the latter point, consider the formula $\langle i\rangle(p \wedge \langle\ell\rangle o) \;\wedge\; \langle j\rangle o \;\wedge\; [j][\ell^{\smile}]\langle i\rangle(p \wedge \langle\ell\rangle o) \wedge \langle j\rangle(o \wedge [\ell^{\smile}]q)$. The first three conjuncts imply the existence of an infinite

(possibly cyclic) R_i-path $w_1, w_2, \ldots$ such that *the* world in which o is true is ℓ-accessible from each w_k. The fourth conjunct implies that, in all w_k, q is true—however, this is only "detected" when the $\Diamond$-rule is applied to the fourth conjunct.

To handle, additionally, nominals, we can further modify our extended $\mathbf{K}_n^{\smile}$ tableau algorithm as follows. Firstly, we give up completion trees. More precisely, if $o_1, \ldots, o_\ell$ are all nominals occurring in ϕ and Γ, we start our tableau with $\ell + 1$ root nodes x_i, where $\mathcal{L}(x_0) = \{\phi\}$ and $\mathcal{L}(x_i) = \{o_i\}$, for each $1 \leq i \leq \ell$. Then, whenever we find a nominal o_i in a node $x \neq x_i$, we *merge* x into x_i; that is, we merge x and x_i's labels and incoming and outgoing edges. As a consequence of this merging, we will possibly find several edges going into a nominal node x_i; however, removing these edges clearly yields a forest structure. Correctness is then straightforward, and termination is due to the fact that (a) each path starting at some x_j is of bounded length because of blocking, and (b) if a successor node was created for some $\Diamond i \psi \in \mathcal{L}(x)$, then we will not create it "again", even if x was merged into another node. For details, see [5, 106].

Further add graded modalities to $\mathbf{K}_n$

In this section, we will discuss, on a rather abstract level, what modifications are necessary to handle graded modalities $\langle i \rangle_n \phi$ and $[i]_n \phi$; for a more detailed description, see [108, 101].

Firstly, following our previous approach, it is quite obvious that, when we find $\langle i \rangle_n \psi \in \mathcal{L}(x)$, we should make sure that we find $n + 1$ i-successors y_j of x with $\psi \in \mathcal{L}(y_j)$. Usually, when we do not find them, we create them all in a single step. Similarly, if we find $[i]_n \psi \in \mathcal{L}(x)$, we must make sure that we do not find more than n i-successors y_j of x with $\psi \in \mathcal{L}(y_j)$. Thus, if there are more such i-successors, we *merge* two of them, say y_j and y_k, i.e., we merge y_k's node label and outgoing edges into y_j's and remove y_k, thus reducing the number of such i-successors by one.

Secondly, in the presence of contradicting graded modalities $\langle i \rangle_n \psi$ and $[i]_{n'} \psi'$ with $\psi \rightarrow \psi'$ and $n' \leq n$ in the label of a node x, the above naive approach would lead to the repeated generation and merging of i-successors of x, and thus to non-termination. To prevent this "yoyo"-effect, when introducing $n+1$ i-successors for some $\langle i \rangle_n \psi \in \mathcal{L}(x)$, we use an explicit inequality relation between these i-successors, do not merge "explicitly unequal" nodes, and extend the notion of a clash to also cover the case where $[i]_n \psi \in \mathcal{L}(x)$ but x has more than n "explicitly unequal" i-successors with ψ in their label.

Thirdly, these modification yield a terminating yet unsound decision procedure: consider, for example, the formula $[i]_1 p \wedge [i]_1 \neg p \wedge \langle i \rangle_2 q$. With the modifications made so far, our tableau algorithm would generate three (explicitly unequal) i-successors y_j of a root node x_0 with $q \in \mathcal{L}(y_j)$, stop, and return "satisfiable", which is clearly the wrong answer. The reason for this incorrect answer is that we only merge surplus i-successors for some $[i]_n \psi$ if we already know that they must satisfy ψ, i.e., if ψ is found in their label. However, as the previous example shows, this is not enough: if $[i]_n \psi \in \mathcal{L}(x)$, we must determine, for each i-successor y_j, whether it does or does not satisfy ψ. We can do this using an additional, non-deterministic *choose*-rule that adds, to each such i-successor, either ψ or the negation normal form of $\neg \psi$.

For $\mathbf{K}_n$, these modifications lead to a decision procedure for satisfiability, even in the presence of either background theories or converse modalities (where we only have to take care to count and merge *i-neighbours* correctly). However, for $\mathbf{K}_n^{\smile}$ with background theories, we need a further modification, namely one to the blocking condition: otherwise, the algorithm is not correct (see, e.g., the example in [105]). Since this logic lacks the finite model property, a construction

of a model from a completion tree uses standard unravelling where, instead of a path going to a blocked node, it goes to the node blocking it. Now, in the presence of graded modalities, we must make sure that this does not lead to additional i-accessible worlds which thus would violate some graded modal formulae. Roughly speaking, we ensure this using *double blocking*, i.e., instead of a node being blocked by an ancestor, a node and its predecessor is blocked by an ancestor and its respective predecessor. For details, see [105, 108].

Implementation Issues

As we have seen, the tableau algorithm $\mathbf{K}_n^{\smile}$ requires a more complex blocking condition in order to ensure that a completion tree can be unravelled into an infinite tableau. This can adversely affect performance, because blocks can take (much) longer to establish, and the completion tree can thus grow (much) larger. The problem can be ameliorated by using a more precise (weaker) blocking condition that identifies the cases where double blocking is really needed (i.e., where a cyclical model cannot be built from a branch of the completion tree blocked using the original single blocking condition), and compares only those parts of the node label pairs that are relevant to determining if the completion tree could be unravelled to give an infinite tableau [107].

Transitive Closure and Fixpoints

There are various extension of modal logics with transitive closure operators and general fixpoints, see Chapter 12 of this handbook. However, there are only few “practicable” satisfiability algorithms in the sense that one could dare to implement them and expect a reasonable behaviour in any non-trivial case.[6] To the best of our knowledge, there are only two such algorithms based on tableau, namely the ones described in [11, 45] for extensions of $\mathbf{K}_n$ with transitive closure, and there has only been a single attempt at an implementation, namely in the system DLP [159]. For this kind of extensions, automata-based techniques (see 5.1) seem to be suited best: for example, the only known decision procedure for the μ-calculus is based on automata, see Chapter 12 of this handbook.

5 OTHER COMPUTATIONAL APPROACHES

5.1 Automata-based algorithms

Roughly speaking, automata-based algorithms work as follows. To decide the satisfiability of a logic $\mathcal{L}$, we first show an appropriate tree-model property for $\mathcal{L}$, i.e., prove that each satisfiable $\mathcal{L}$ formula is satisfiable in a model (or an abstraction of a model) whose relational structure forms a tree. For example, it is well-known that each satisfiable $\mathbf{K}_n$ formula is satisfiable in a tree model which is, additionally, finite [93]. For other logics, e.g., $\mathbf{K4}_n$, we can easily show that each satisfiable formula has a model with an infinite tree *abstraction*, where we can obtain a model from such an abstraction by transitively closing the accessibility relations [93]. Secondly, for an $\mathcal{L}$ formula ϕ, we define an automaton $\mathcal{A}_\phi$ such that $\mathcal{A}_\phi$ accepts all tree models of ϕ (or abstractions thereof). Depending on the logic and its model properties, we use automata on finite or on infinite trees. Thus we have reduced the satisfiability of formulae in $\mathcal{L}$ to the emptiness

[6] For other reasoning problems such as model checking, these algorithms exist and have been implemented successfully, see Chapter 17 of this handbook.

problem of a certain class of automata, and we can use well-known algorithms to decide these emptiness problems.

For a variety of logics, this approach has several of advantages. Consider, for example, $\mathbf{K}_n$ with background theories. It can easily be seen that this logic enjoys the tree model property, and thus we only need to devise the construction of an automaton $\mathcal{A}_\phi$. Using *alternating automata*, this construction is quite straightforward and yields, surprisingly, a (worst-case) optimal decision procedure (for a similar construction for a more powerful logic see, e.g., [188]): the automaton $\mathcal{A}_\phi$ is of size polynomial in the input tree, and testing its emptiness can be done in deterministic exponential time [128]. Thus, in contrast to the tableau algorithm described in Section 4.4, we effortlessly obtain a *deterministic* algorithm, and do not even need to take care of termination or finite models: using automata on infinite trees makes this unnecessary.

Concerning the implementability of automata-based approaches, we observe that their worst-case complexity often coincides with their best-case complexity: to decide the emptiness of alternating automata, we first translate them into non-deterministic ones that are then tested for emptiness, i.e., we first build a structure of exponential size, for which we then decide emptiness in polynomial time [128]. In case we directly use non-deterministic automata, they tend to be of size exponential in the size of the input formula, and we are thus confronted with the same problem. Thus, any naive implementation is doomed to failure. However, there are at least two ways out: in [158], it was shown how BDDs can be used to efficiently represent and handle large automata, thus proving that (variations of) automata-based algorithms can be implemented efficiently using appropriate data structures. In [12], it was shown how an automata-based approach can be transformed mechanically into a tableau-based decision procedure: as a consequence, we only need to "hand-craft" the automata-based algorithm, and then get both a (possibly optimal) worst-case upper bound and a (possibly practicable) tableau-based algorithm for free.

5.2 Modal resolution

In the late 1980s and early 1990s various direct resolution methods for modal logics have been investigated [1, 10, 33, 46, 59, 61, 72, 79, 126, 139, 140]. According to [139] a *resolution method* for a logic L is determined by specifying (i) a class of formulae called clauses, (ii) a reduction method which allows us to transform any formula of L into a finite set of clauses, (iii) a calculus consisting of a set of resolution rules for deriving clauses (and possibly redundancy elimination and simplification rules), and (iv) a derivation process which starts from an initial set of clauses and constructs a sequence of derivable clauses. One can then define a *modal resolution method* to be a resolution method in which clauses are formulae of the modal logic L under consideration. This definition excludes methods which do not use a clausal form from the outset, e.g. destructive modal resolution [72], or methods which use auxiliary labels, e.g. prefixed resolution [6] and labelled modal resolution [7]. Methods which use additional modal operators like the resolution calculus for temporal logics of knowledge presented in [54] can be considered to be borderline cases.

In the following we focus on the modal resolution method of [59] but follow the presentation in the survey paper [64], where a more complete overview of various direct resolution methods and other methods can be found.

A modal formula of $\mathbf{K}$ is in *disjunctive normal form* iff it is a (possibly empty) disjunction of the form $\bigvee L_i \vee \bigvee \Box D_j \vee \bigvee \Diamond A_k$ where each L_i is a propositional literal, each D_j is a modal formula in disjunctive normal form, and each A_k is a modal formula in conjunctive normal form. A modal formula is in *conjunctive normal form* iff it is a conjunction $\bigwedge D_l$ where each

Axioms			
axiom1:	$p, \neg p \Rightarrow \bot$	axiom2:	$\bot, A \Rightarrow \bot$

Resolution rules		
$\vee$-rule1:	$A \vee D, B \vee D' \Rightarrow C \vee D \vee D'$	if $A, B \Rightarrow C$
$\vee$-rule2:	$A \vee C \Rightarrow B \vee C$	if $A \Rightarrow B$
$\Diamond$-rule1:	$\Diamond(A, B, N) \Rightarrow \Diamond(A, B, C, N)$	if $A, B \Rightarrow C$
$\Diamond$-rule2:	$\Diamond(A, N) \Rightarrow \Diamond(B, A, N)$	if $A \Rightarrow B$
K-rule1:	$\Box A, \Diamond(B, N) \Rightarrow \Diamond(B, C, N)$	if $A, B \Rightarrow C$
K-rule2:	$\Box A, \Box B \Rightarrow \Box C$	if $A, B \Rightarrow C$
$\Box$-rule:	$\Box A \Rightarrow \Box B$	if $A \Rightarrow B$

Simplification rules			
$\vee$-simp1:	$\bot \vee D \rightarrow D$	$\Diamond$-simp:	$\Diamond\bot \rightarrow \bot$
$\vee$-simp2:	$A \vee A \vee D \rightarrow A \vee D$	$\wedge$-simp:	$\bot, N \rightarrow \bot$

Figure 15. Modal resolution rules of [59] for $\mathbf{K}$. (The symbols A, B, C, D, D' denote clauses, N denotes a set of clauses, and (A, N) denotes the union of $\{A\}$ and N. No distinction is made between a set N of clauses and the conjunction of its elements.)

D_l is a modal formula in disjunctive normal form. A formula in disjunctive normal form is also called a *(modal) clause*. Any modal formula φ can be transformed into an equivalent formula in conjunctive normal form $\mathrm{cnf}(\varphi)$. In the following, we do not distinguish between a conjunction of clauses and a set of clauses.

The calculus $C_{\mathbf{K}}$ of [59] is given by the set of axioms, resolution rules, and simplification rules shown in Figure 15. The intended meaning of $A, B \Rightarrow C$ and $A \Rightarrow C$ is that the conjunction of the formulae on the left-hand side of $\Rightarrow$ implies the formula on its right-hand side. In contrast, the meaning of $A, B \rightarrow C$ is that occurrences of A and B in a conjunction can be simplified to, that is, replaced by, C. Analogously, $A \rightarrow C$, means that occurrences of A can be replaced by C. Every formula A has a unique normal form $\mathrm{nf}(A)$ under the simplification rules of Figure 15 (modulo commutativity and associativity of $\vee$ and $\wedge$).

Various extensions of $\mathbf{K}$ have been considered, including extensions by the axiom schemas $\mathbf{D}$, $\mathbf{T}$, and $\mathbf{4}$. For each of these axiom schemas the calculus $C_{\mathbf{K}}$ needs to be extended with additional rules: for $\mathbf{D}$ with $\Box\bot \Rightarrow \bot$, for $\mathbf{T}$ with $\Box A, B \Rightarrow C$ if $A, B \Rightarrow C$, while for $\mathbf{4}$ with the two rules $\Box A, \Box B \Rightarrow \Box C$ if $\Box A, B \Rightarrow C$ and $\Box A, \Diamond(B, N) \Rightarrow \Diamond(B, C, N)$ if $\Box A, B \Rightarrow C$. We denote the calculi obtained by adding these rules to $C_{\mathbf{K}}$ by $C_{\mathbf{KD}}$, $C_{\mathbf{KT}}$, and $C_{\mathbf{K4}}$, respectively.

Let L be one of $\mathbf{K}$, $\mathbf{KD}$, $\mathbf{KT}$, $\mathbf{K4}$. Given sets of clauses N and (C, N) we say (C, N) can be *derived in one step* from N in C_L iff either there are clauses A and B in N such that $A, B \Rightarrow C'$ in C_L or there is a clause A in N such that $A \Rightarrow C'$ in C_L, and $C = \mathrm{nf}(C')$ in C_L. A *derivation* of N' from N in C_L is a sequence $N = N_0, N_1, \ldots, N_n = N'$ such that for every i, $0 \leq i < n$, N_{i+1} can be derived from N_i in one step. A *refutation* of N in C_L is a derivation of $\bot$ from N in C_L. If a refutation of N exists, then N is C_L*-refutable*.

In [59] it is shown that a modal formula φ is valid in L iff $\mathrm{cnf}(\neg\varphi)$ is C_L-refutable. This soundness and completeness result is shown in [10] to also hold for a number of refinements of this modal resolution method and the extension by subsumption deletion.

So far, little work seems to have been conducted on devising specialised and efficient data structures and algorithms for modal resolution methods. Due to the extra structural information

that modal formulae carry, which is reflected in the more complicated clausal form, the data structures and algorithms developed for efficient propositional and first-order resolution provers cannot be utilised easily to implement modal resolution methods.

5.3 Sequent-based approaches

Sequent calculi were introduced by Gentzen [80] as a tool for studying natural deduction. The central property of sequent calculi is cut elimination which usually yields consistency as an easy corollary. The first sequent calculi and cut elimination results for modal logics have been established in the early fifties [39], see [89] for further historic references.

A *sequent* is a structure of the form $\Gamma \vdash \Delta$, where Γ and Δ are (finite) lists, multisets, or sets of formulae; Δ is also quite often restricted to be a singleton set or the empty set. A *sequent calculus* for a logic L consists of two parts: (i) a finite set of axioms, (ii) a finite set of rules of the form $\frac{S_1}{S}$ or $\frac{S_1\ S_2}{S}$ with conclusion S and premises S_1 and S_2, where S, S_1, and S_2 denote sequents. The rules can usually be divided into two major groups: *logical rules*, which introduce a new logical formula either on the left or on the right of the turnstile $\vdash$, and *structural rules*, which operate on the structure of the sequents. Of particular interest, both from a proof-theoretical and a computational point of view is the *cut rule*, a rule of the form

$$\frac{\Gamma_1 \vdash \Delta_1, A \qquad A, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}$$

where, in general, A is an arbitrary formula, called the *cut formula*. A *sequent calculus proof* of a *goal sequent* S is a tree whose nodes are labelled with sequents, such that (i) the root of the tree is labelled with S, (ii) each leaf node is an instance of an axiom of the calculus, and (iii) each sequent labelling a non-leaf node n follows by one of the rules of the calculus from the sequents labelling the children of n. This notion of a proof does not prescribe a particular approach to the construction of the proof of a sequent S. However, it is quite natural to proceed by *backward reasoning*, that is, to start with a tree consisting only of the root node labelled with S and to apply rules from bottom to top, taking the sequent labelling a node of the tree to be the conclusion of a rule and adding children to the tree labelled with the premises of the rule. The construction is complete if all the current leaf nodes are labelled with instances of axioms. In contrast, in *forward reasoning* one would start with one or more leaf nodes labelled with instances of axioms and build the tree toward its root node labelled with S. This approach is basically taken in the *inverse method*, see Section 5.4. For further details on sequent calculi see Sections 7 and 8 of Chapter 2 of this handbook.

From a computational point of view, sequent calculi pose several challenges and also provide insights that can help to improve systems based on tableau calculi or the inverse method.

First, the cut rule is problematic for backward reasoning, since we can choose an arbitrary formula to be the cut formula. We can try to show that we can restrict ourselves to cut formulae which are subformulae of formulae in the goal sequent S while retaining completeness of the calculus. The result would be a calculus with *analytic cut*. However, from a practical point of view, while for analytic cuts we can only choose finitely many different cut formulae, the search space may still be too large. Alternatively, one can try to show that for any sequent S there exists a proof without any application of the cut rule. In such a case, we can omit the cut rule from the calculus and obtain a *cut-free sequent calculus*. While for some modal logics it is rather straightforward to devise cut-free sequent calculi, for others it is much more challenging, for example, for **S5** [30, 151, 152], and for some it is an open problem, for example, for **PDL**

and converse **PDL** [127]. We are also not aware of cut-free systems for modal logics with the common knowledge operator.

Second, in the presence of the axiom schema **4**, systems based on sequent calculi face the same non-termination problems as systems based on tableau calculi. Recall from Section 7 of Chapter 2 one of the additional rules required for **K4**,

$$\frac{\Box Y_1, \ldots, \Box Y_m, Y_1, \ldots, Y_m, A \vdash \Diamond Z_1, \ldots, \Diamond Z_n, Z_1, \ldots, Z_n}{\Gamma, \Box Y_1, \ldots, \Box Y_m, \Diamond A \vdash \Diamond Z_1, \ldots, \Diamond Z_n, \Delta}$$

where Γ and Δ are sequences of formulae not containing a $\Box$-formula and $\Diamond$-formula, respectively. Here, the premise is not necessarily 'simpler' than the conclusion which can lead to situations in which this rule can be applied infinitely many times when using backward reasoning. To ensure termination a form of *loop-check* has to be used, that is, a check which detects whenever the 'same' sequent occurs twice on a branch of a proof. If in turn we would like to formulate the rules of our calculus in such a way that the applicability of rules does not depend on information about the whole branch of a proof or even the whole proof, additional *history information* has to accompany each sequent in a proof. What minimal history information for loop-checks is necessary to ensure termination on a variety of modal logics, including **KT** and **S4**, is investigated in [96, 98]. These results transfer directly to tableau calculi.

Finally, sequent-based systems face the same problems as tableau-based systems when trying to prove formulae involving disjunctions on the left or conjunctions on the right of the turnstile. Naturally, similar solutions as presented in Section 4, in particular, simplification and forms of intelligent backtracking, have also been considered in the context of sequent-based systems, most notably in the work of [25, 96, 97].

5.4 *Inverse method*

The inverse method is a variant of the sequent calculus [51, 135] which carries its name because it works from sub-goals to goals, whereas standard sequent-based approaches work in the other direction. For example, if it has already been proven that ϕ is false and ψ is true, then the inverse method will deduce from this that $\phi \to \psi$ is true. For this kind of forward reasoning to work, we need to be able to focus on an acceptably small set of axioms, and an acceptably small set of goals and sub-goals. For many modal logics, we can restrict our attention to such 'acceptably small' sets of formulae since they enjoy the sub-formula property, i.e. every valid formula ϕ has a derivation in which only (negated or unnegated) sub-formulae of ϕ occur [51]. Calculi for modal logics using the inverse method have been developed in [139, 140, 190]. The inverse method has been shown to be suitable for efficient modal logic theorem proving and is amenable to optimisations [190].

Interestingly, the inverse method is closely related to automata-based approaches [13]. More precisely, the algorithm that decides emptiness of automata (the problem to which satisfiability of a variety of modal logics can be reduced, see Section 5.1) can be viewed as being a notational variant of the inverse method. Both start with propositional axioms (in the automata emptiness test, these correspond to unreachable states), and saturate these axioms using basically the same deduction rules. As a consequence, it should be possible to translate a variety of automata-based decision procedures into the inverse method, thus obtaining an efficient implementation (or a good starting point for its implementation) basically for free.

6 OTHER REASONING PROBLEMS

In this chapter, we have focused on one specific reasoning problem, satisfiability or, dually, validity. There are, however, other interesting reasoning problems for modal logics that are useful for certain applications. We will discuss some of them in this section and we refer the reader to Section 5 of Chapter 13 of this handbook for reasoning problems that are motivated by applications of description logics.

6.1 Model checking

Model checking is the problem of deciding whether $\mathfrak{M}, w \models \varphi$ for a given Kripke structure $\mathfrak{M}$, a world w, and a modal formula φ. It is used for system verification, e.g. to verify a piece of soft- or hardware, as follows:

- $\mathfrak{M}$ represents the system: worlds are viewed as *states* the systems can be in,
- modal parameters represent actions which take the system from one state into another (or several others),
- w is some initial state, and
- φ is a temporal logic formula describing a desired behaviour of the system.

Whereas satisfiability algorithms have to reason w.r.t. *all* structures (possibly from a given class), model checking is concerned with a single structure, and thus quite different: model checking is often less complex than satisfiability, and there are industrial strength implementations of model checking algorithms capable of handling large systems and formulae from rather expressive logics. We refer the interested reader to Chapter 17 of this handbook and [34, 35].

6.2 Proof checking

Proof checking is the problem of deciding whether a given derivation P is a proof of a given formula φ, commonly with respect to a fixed calculus C for a logic L. It requires a language in which we are able to formalise derivations. The formalisation of a derivation may simply be a sequence or a tree-structure of formulae, but may also contain additional information about which and how inference rules of the calculus C have been used in each step of the derivation.

The motivation for proof checking is the fact that advanced theorem proving systems are rarely verified. Thus, like any other piece of software they invariably include errors which can lead the system to provide incorrect answers, including, providing an incorrect proof P for a given formula φ. Simplifying theorem proving systems to an extend that would allow their verification in all likelihood results in systems which are too slow to be useful. However, such system may still be sufficiently powerful to check the correctness of a given derivation P. Thus, a natural approach is to use a highly optimised but unverified system to find a proof P for a given formula φ which is then independently checked for correctness by a slower, verified system.

Proof checking has received considerable attention in the context of higher-order logic [155, 195] and is taken seriously in the context of first-order logic [136]. However, we are not aware of any work in this direction in the context of modal logics, although the problem of incorrect theorem provers also exists in this field. Note that in the context of the translation approach we can rely on first-order proof checkers augmented with a verified program for translating modal formulae into first-order clause sets.

Even more complex is the problem of verifying the non-existence of a proof. For the modal logics we have considered in this chapter we would also expect decision procedures to correctly determine in finite time that a given formula φ has no proof. A justification for this can be given by a model or representation of a model $\mathfrak{M}$ for $\neg\varphi$, produced by the decision procedure. A verified model checker could then be used to independently verify that $\mathfrak{M}$ is indeed a model of $\neg\varphi$.

6.3 Computing correspondences

Recall from Chapter 1 of this handbook the notion of a modal formula $\varphi(p_1, \ldots, p_n)$ over propositional variables $p_1, \ldots, p_n$ being *true in frame* $\mathfrak{F}$ iff for every world w and every valuation mapping V for its propositional variables we have $(\mathfrak{F}, V), w \models \varphi$, and the notion of a modal formula φ defining a class of frames iff φ is true in precisely the frames in the class. It straightforward to see that a modal formula φ over $p_1, \ldots, p_n$ is true in a frame $\mathfrak{F}$ iff the monadic second-order formula $\forall P_{p_1} \ldots P_{p_n} \forall x\, \pi_r(\varphi, x)$ is true in the class of all models over the frame $\mathfrak{F}$. There are methods for reducing such second-order formulae to equivalent first-order formulae and there are methods for reducing the second-order logic formulation of modal axioms to the corresponding frame properties. Computing the first-order equivalents of modal formulae (if they exists) amounts to the elimination of the universal or existential monadic second-order quantifiers. For example, if we are interested in establishing the *relational frame properties* corresponding to a modal formula φ, then we either have to eliminate the universal monadic second-order quantifiers from $\forall P_{p_1} \ldots P_{p_n} \forall x\, \pi_r(\varphi, x)$, or, equivalently, the existential monadic second-order quantifiers from $\exists P_{p_1} \ldots P_{p_n} \exists x\, \pi_r(\neg\varphi, x)$. There can be no algorithm which is guaranteed to find a first-order equivalent formula if there exists one. Still, a number of automated algorithms are known which provide a partial solution to the quantifier elimination problem, namely SCAN [75, 58], DLS [55, 185] and SQEMA [37]. SCAN and DLS are based on a form of resolution while SQEMA can be viewed as a modalized DLS algorithm. Here we briefly review the SCAN algorithm, but more details of DLS and other quantifier elimination algorithms can be found in [36, 147].

The SCAN algorithm involves three stages:

(i) transformation to clausal form and (inner) Skolemisation;

(ii) C-resolution;

(iii) reverse Skolemisation (unskolemisation).

The input of SCAN is a second-order formula of the form $\exists Q_1 \ldots \exists Q_k\, \psi$, where the Q_i are unary predicate variables and ψ is a first-order formula. In the first stage SCAN converts ψ into clausal normal form by transformation into conjunctive normal form, Skolemisation, and clausifying the Skolemised formula. In the second stage SCAN performs a special kind of constraint resolution, called *C-resolution*, the two main inference rules are given in Figure 16. It generates all and only resolvents and factors with the second-order variables that are to be eliminated, which in the case of computing frame correspondence properties includes all existentially quantified second-order variables. When all C-resolvents and C-factors with respect to a particular Q_i-literal and the rest of the clause set have been generated, purity deletion removes all clauses in which this literal occurs. The subsumption deletion rule is optional for the sake of soundness, but helps simplify clause sets in the derivation.

If the C-resolution stage terminates, it yields a set N of clauses in which the specified second-order variables are eliminated. This set is satisfiability equivalent to the original second-order

formula. If no clauses remain after purity deletion, then the original formula is a tautology; if C-resolution produces the empty clause, then it is unsatisfiable. If N is non-empty, finite and does not contain the empty clause, then in the third stage, SCAN attempts to restore the quantifiers from the Skolem functions by reversing Skolemisation. This is not always possible, for instance if the input formula is not first-order definable.

If the input formula is not first-order definable and stage two terminates successfully yielding a non-empty set not containing the empty clause then SCAN produces equivalent second-order formulae in which the specified second-order variables are eliminated but quantifiers involving Skolem functions occur and the reverse Skolemisation typically produces Henkin quantifiers. If SCAN terminates and reverse Skolemisation is successful, then the result is a first-order formula logically equivalent to the second-order input formula.

SCAN can compute the frame correspondence properties for very many well-known axioms including $\mathbf{T}$, $\mathbf{4}$, and $\mathbf{5}$. Recent work has in fact shown that the SCAN algorithm is complete for the class of all Sahlqvist formulae, in the sense that, when given a Sahlqvist formula it will successfully compute an equivalent first-order formula for it [88].

6.4 Model generation

A problem closely related to the satisfiability problem is the problem of generating (counter-) models. Ideally we want to construct finite models if they exist. It is possible to use both tableau and resolution methods to prove that logics have the finite model property and also to give procedures for constructing standard Kripke models.

Although tableau provers do not always output models, it is well-known that tableau procedures implicitly generate models (of some kind) for satisfiable input problems. This is especially true for semantic tableau procedures which are defined by structural rules and use explicit accessibility relations. Modal tableau procedures of the kind described in Section 4 which use propagation rules for handling the additional axioms do construct models but often they are just skeleton models which need to be completed with respect to the relational correspondence properties and then give standard Kripke models.

In first-order logic it is well-known that hyperresolution like tableau methods can be employed both as a reasoning method and a Herbrand model builder [31, 65]. It has been shown that the methods using $\mathsf{R}^{\mathsf{hyp}}$ and the relational translation described in Section 3 require hardly any extra effort to construct a modal model [49, 121, 180]. It is usually a simple matter to read off a Kripke model from the saturated set of ground unit clauses which represents a Herbrand model. In general this set will be infinite in the limit, but when $\mathsf{R}^{\mathsf{hyp}}$ is a decision procedure then the set is finitely bounded and consequently a finite Kripke model can be defined.

In more detail, a *Herbrand interpretation* is a set of ground atoms. By definition a ground atom A is *true* in an interpretation H iff $A \in H$ and it is *false* in H iff $A \notin H$. Now, extend the

C-Resolution: $$\frac{C \vee Q(s_1, \ldots, s_n) \quad \neg Q(t_1, \ldots, t_n) \vee D}{C \vee D \vee s_1 \not\approx t_1 \vee \ldots \vee s_n \not\approx t_n}$$
provided the two premises have no variables in common and are distinct clauses

C-Factoring: $$\frac{C \vee Q(s_1, \ldots, s_n) \vee Q(t_1, \ldots, t_n)}{C \vee Q(s_1, \ldots, s_n) \vee s_1 \not\approx t_1 \vee \ldots \vee s_n \not\approx t_n}$$

Figure 16. The calculus of SCAN

definition as expected to the Boolean combination of ground atoms. A clause C is true in H iff for all ground substitutions σ there is a literal L in $C\sigma$ which is true in H. A set N of clauses is true in H iff all clauses in N are true in H. If a set N of clauses is true in an interpretation H then H is referred to as a *Herbrand model* of N. It is proved in [49, 121] that the combination of the relational translation and R^{hyp} can be used as a finite Herbrand model generator for the modal logics $\mathbf{K}_n$, $\mathbf{K}_n^{\smile}$ and the extensions with $\mathbf{T}$, $\mathbf{D}$, $\mathbf{B}$ (actually more general results are proved).

In general Herbrand models are not unique and can be large. Therefore it is useful to have a method for generating minimal Herbrand models. An interpretation H is a *minimal Herbrand model* for a set N of clauses iff H is a Herbrand model of N and for no Herbrand model H' of N, $H' \subset H$ holds. Various approaches to generating minimal Herbrand models with hyperresolution are known [26, 31, 94, 144]. It follows from [31] and investigations of $\mathbf{GF}^-$ and the class $\mathcal{BU}$ in [81, 82] that with a moderate extension of R^{hyp}, denoted here by $R^{\mathsf{hyp}}_{\mathsf{min}}$, it is possible to guarantee the generation of all and only minimal Herbrand models for any modal and description logic reducible to a decidable class of range restricted clauses. It is necessary to use a depth-first strategy, a complement splitting rule should be used so that the first model generated is a minimal Herbrand model, and a model constraint propagation rule is necessary to prevent the generation of non-minimal Herbrand models (see Figure 17). The procedure $R^{\mathsf{hyp}}_{\mathsf{min}}$ is generally sound and complete and is a minimal Herbrand model building procedure for range-restricted clauses [31]. An alternative is to use the generalisation [81, 82] of an approach of [144].

It is not difficult to see that model generation procedures and the mentioned minimal Herbrand model generation procedures can be developed by using hyperresolution and the other translation methods. Because of the close connection to tableau, corresponding tableau procedures can be defined and all results carry over to the tableau setting (see [49, 121]).

6.5 Bisimulation

Chapter 1 of this handbook has introduced the notion of a bisimulation between two Kripke models. A *bisimulation* between models $\mathfrak{M} = \langle W, R, V\rangle$ and $\mathfrak{M}' = \langle W', R', V'\rangle$ is a binary relation $E \subseteq W \times W'$ such that whenever $E(w, w')$ the following three properties hold:

Atomic: for all propositional variables p, $w \in V(p)$ iff $w \in V'(p)$;

Zig: if $R_i(w, v)$ for some i, then there exists v' in $\mathfrak{M}'$ such that $E(v, v')$ and $R'_i(w', v')$; and

Zag: if $R'_i(w', v')$ for some i, then there exists v in $\mathfrak{M}$ such that $E(v, v')$ and $R_i(w, v)$.

Complement splitting: $$\frac{N \cup \{C \vee D\}}{N \cup \{C, \neg D\} \quad | \quad N \cup \{D\}}$$

where D is a ground clause.

Model constraint propagation: $$\frac{N}{N \cup \{\neg A_1 \vee \ldots \vee \neg A_n\}}$$

where $\{A_1, \ldots, A_n\}$ is the finite Herbrand model of an open branch which is complete with respect to R^{hyp}. The model constraint propagation rule extends all branches in the derivation tree (to the right) which are not complete with respect to $R^{\mathsf{hyp}}_{\mathsf{min}}$.

Figure 17. Additional rules for minimal Herbrand model generation

One important property of bisimilar models is that they satisfy the same μ-calculus formulae, that is, let $E(w, w')$ hold then a μ-calculus formula φ is true at w in $\mathfrak{M}$ iff φ is true at w' in $\mathfrak{M}'$. The notion of bisimulation not only plays an important rôle in modal logic, as an equivalence principle between Kripke models, but also in other fields, for example, concurrency theory, set theory, and formal verification. An algorithm for 'on the fly' verification of bisimulations is presented in [66].

A related problem is that of *bisimulation minimisation*, that is, the problem of finding the minimal Kripke model bisimilar to a given Kripke model. In particular, in the context of formal verification by model checking (see Section 6.1 and Chapter 17 of this handbook for further details), bisimulation minimisation provides an easily and automatically computable way to reduce the number of states of a model while preserving the truth and falsehood of the formulae that hold in it.

Let $\mathfrak{M} = \langle W, R, V\rangle$ be a Kripke model and E be an equivalence relation on W. Let $[w]_E$ denote the equivalence class of a world $w \in W$ with respect to E. The set of all equivalence classes is a *partition* of W and $\mathfrak{M}$. The bisimulation minimisation of a Kripke model $\mathfrak{M}$ is the quotient $\mathfrak{M}/E = \langle W', R', V'\rangle$ where

$$\begin{aligned} W' &= \{[w]_E \mid w \in W\}, \\ R' &= \{([w]_E, [w']_E) \mid w, w' \in W \wedge R'(w, w')\}, \text{ and} \\ V'(p) &= \{[w]_E \mid w \in V(p)\} \end{aligned}$$

for every propositional variable p such that E is the maximal equivalence relation on W which is also a bisimulation between $\mathfrak{M}$ and itself. A partition P is *stable* with respect to E iff for each pair $[w]_E$, $[w']_E$ of equivalence classes with respect to E either $[w]_E \subseteq E^{-1}([w']_E)$ or $[w]_E \cap E^{-1}([w']_E) = \emptyset$.

In the computation of the bisimulation minimisation of a Kripke model we can basically follow two strategies. One is a negative strategy in which we start with the coarsest partition P such that $E(w, w')$ iff $w \in V(p)$ iff $w' \in V(p)$ for every positional variable p and split classes whenever P is not stable. Another is a positive strategy in which we start with the finest partition P in which each equivalence class consists of a single world and the bisimulation minimisation is constructed via a sequence of steps in which we merge two or more classes. An algorithm following the negative strategy is presented in [156] which has the optimal worst-case running time, namely $O(|R| log |W|)$. An implementation of this algorithm is presented in [67]. Other algorithms following a negative strategy are presented in [28, 130]. They take advantage of the fact that in a number of applications we are only interested in the part of a Kripke model reachable from a designated start world. In this case, equivalence classes associated with unreachable worlds need not be taken into account when considering the stability of an equivalence class associated with a reachable world. An algorithm following the positive strategy is presented in [157]. Recently, [57] has introduced an algorithm combines both the positive and negative strategy by using the algorithms of [156] and [157] as subroutines. For a range of special cases this algorithm terminates in time $O(|R| + |W|)$.

Finally, [71] presents on-the-fly model checkers for invariant properties incorporating the bisimulation minimisation algorithms of [28, 130, 156]. From an empirical comparison they draw the conclusion that in this context an optimised version of the algorithm of [156] performs better than the other two.

6.6 Modal logic programming

The problem of extending logic programming languages with modal operators has received a lot of attention in the late 1980s and early 1990s, at about the same time most of the direct resolution methods mentioned in Section 5.2 were developed and also work on the translation methods described in Section 3 intensified. Consequently, work in this area can again be divided between direct approaches and translation approaches.

Following the direct approach, [21, 20] presents a declarative semantics and an SLD resolution calculus for a class of modal logic programs in modal logics $\mathbf{KD}$, $\mathbf{KT}$, and $\mathbf{S4}$, while [22, 23] present a framework for developing the fixpoint and operational semantics of a class of multi-modal logic programs where additional properties of modal operators can be described by axiom schemas of the form $[i_1][i_2]\cdots[i_m]p \rightarrow [j_1][j_2]\cdots[j_n]p$, so-called inclusion axioms. More recent work includes [142] presenting a fixpoint semantics, least model semantics, and an SLD resolution calculus for modal logic programs in modal logics extending $\mathbf{K}$ with a non-empty selection of the axiom schemas $\mathbf{B}$, $\mathbf{D}$, $\mathbf{T}$, $\mathbf{4}$ and $\mathbf{5}$. Also, modal logic programs in [142] are as expressive as the general modal Horn fragment which allows arbitrary occurrences of the modal operators $\Box$ and $\Diamond$ in programs clauses and goals.

Following the translation approach, [50] applies the functional translation to multi-modal logic programs in the modal logics $\mathbf{KD}$, $\mathbf{KT}$, $\mathbf{KD4}$, $\mathbf{KT4}$, $\mathbf{KF}$ ($\mathbf{F}$ is the functionality axiom), and simple inclusion axioms of the form $[i]p \rightarrow [j]p$. In these logics, the functional translation of goals and program clauses in the general modal Horn fragment are in the first-order Horn fragment. For computations SLD resolution extended by theory unification is used. In [146] presents an application of the *semi-functional translation* [145, 148] to modal logic programs in modal logics $\mathbf{KB}$ and $\mathbf{KDB}$, as well as $\mathbf{KD}$, $\mathbf{KT}$, and their extension by one or both of the axiom schemas $\mathbf{4}$ and $\mathbf{5}$. The semi-functional translation combines features of the relational and functional translation. For modal formulae in negation normal form, subformulae of the form $\Box\varphi$ are translated using the relational translation, while subformulae of the form $\Diamond\varphi$ are translated using the functional translation. A *functional simulator axiom* needs to be added to the translation to link the relational and functional aspects of the translation. The semi-functional translation has the advantage over the functional translation that the frame properties of many modal logics, including the ones listed above, can be specified by simple first-order Horn theories without equality. Consequently, the use of theory unification and theory resolution can be avoided. Furthermore, if the semi-functional translation is applied to goals and program clauses in the general modal Horn fragment, then the resulting first-order clauses are themselves Horn. Together with the fact that the frame properties are expressed by Horn clauses, this implies that unmodified SLD resolution can be used to execute the translated modal logic programs.

The functional and semi-functional translation have been incorporated into MSPASS [120, 174]. Implementations of systems based on the direct approach include MOLOG [62, 63], MProlog [141, 143], and TIM [21]. However, just as for the direct resolution approaches described in Section 5.2, little work seems to have been conducted on developing specialised and efficient data structures and algorithms for such systems, with the exception of [2] which describes an abstract machine model for MOLOG, in analogue to the Warren Abstract Machine model for Prolog [191].

There has also been considerable work on temporal logic programming. For surveys on this work which also cover some of the approaches to modal logic programming mentioned above see [154, 70, 84].

There is currently renewed interest in modal and temporal logic programming in the context of multi-agent system development [24, 53, 69] and related areas.

7 REVIEW AND DISCUSSION

In this chapter we have examined computational approaches to modal logics. Although we have considered a variety of computational approaches and reasoning problems, we have focused on the use of translation-based and tableau-based algorithms for deciding the satisfiability of a formula, both with and without reference to a background theory. This focus was motivated by the dominance of translation-based and tableau-based approaches in implemented systems, and by the importance of satisfiability testing in applications such as the verification of multi-agent systems and ontology engineering.

The reason for the dominance of these two approaches is that they have proved amenable to implementation and optimisation techniques that dramatically improve typical case performance; the use of such techniques is crucial if reasoning systems are to be effective in applications. The applicability and effectiveness of optimisation techniques and refinements is, however, highly dependent on the logic under consideration and on the class of problem being solved. For example, in the context of tableau-based algorithms, caching must be used with care in the presence of converse modalities, and semantic branching search, while highly effective for randomly generated problems, may be ineffective (and perhaps even counter productive) for problems derived from ontology engineering applications. Similarly, in the context of translation-based algorithms, hyperresolution may be the most suitable approach for randomly generated problems in the modal logic $\mathbf{K}_n$, while ordered resolution is more effective for problems derived from ontology engineering applications.

Regarding the two approaches, both have advantages and disadvantages. Tableau-based methods generally require full implementation, but this allows the implementor to choose and fine-tune the optimisations, data structures, and algorithms for effective operation in the intended application. In contrast, no major implementation effort is needed for translation-based methods, but a careful choice of translation, refinement of resolution, and operational parameters is required to guarantee termination and effectiveness of the first-order logic prover on the class of problems being solved. The choice of approach may ultimately depend on the logic in question: tableau-based methods seem to have some advantages in the presence of graded modalities (counting), for example, whereas translation-based methods can handle and may be better for boolean modal logics (role negation). Currently, tableau-based approaches are the most widely used in ontology applications, with description logic systems such as FaCT++, Racer and Pellet [159, 90, 160]. In contrast, translation-based methods have a number of other uses, for example, computing correspondence properties and modal logic programming.

The use of implemented systems in realistic applications brings with it new challenges, both with respect to the expressive power of the logics being used, and the size and complexity of the problems to be solved. The W3C standard ontology language OWL, for example, corresponds to a logic with transitive, converse and graded modalities, as well as nominals, and ontology applications may call for reasoning with respect to very large background theories. For the logic corresponding to OWL, a tableau-based algorithm has only recently been introduced [110], a translation-based algorithm using the basic superposition calculus is still under development [112], and the development of computational and optimisation techniques is the subject of considerable ongoing research. Similarly, as mentioned in the introduction, agent frameworks consist of complex multi-modal logics, typically including a dynamic component, allowing the representation of dynamic activity via a temporal or a dynamic logic. Ongoing research is focusing on developing advanced computational methods and optimisation techniques for such frameworks.

BIBLIOGRAPHY

[1] M. Abadi and Z. Manna. Modal theorem proving. In J. H. Siekmann, editor, *Proceedings of the 8th International Conference on Automated Deduction (CADE-8)*, volume 230 of *Lecture Notes in Computer Science*, pages 172–189. Springer, 1986.

[2] J.-M. Alliot, A. Herzig, and M. Lima Marques. Implementing Prolog extensions: a parallel inference machine. In *Proceedings of the International Conference on Fifth Generation Computer Systems '92*, pages 833–842. IOS Press, 1992.

[3] H. Andréka, I. Németi, and J. van Benthem. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.

[4] C. Areces. *Logic Engineering. The Case of Description and Hybrid Logics*. PhD thesis, ILLC, University of Amsterdam, 2000.

[5] C. Areces, P. Blackburn, and M. Marx. A road-map on complexity for hybrid logics. In *Annual Conference of the European Association for Computer Science Logic (CSL '99)*, volume 1683 of *Lecture Notes in Computer Science*, pages 307–321. Springer, 1999.

[6] C. Areces, H. De Nivelle, and M. de Rijke. Prefixed resolution: A resolution method for modal and description logics. In H. Ganzinger, editor, *Proceedings of the 16th International Conference on Automated Deduction (CADE-16)*, volume 1632 of *Lecture Notes in Artificial Intelligence*, pages 187–201. Springer, 1999.

[7] C. Areces, M. de Rijke, and H. De Nivelle. Resolution in modal, description and hybrid logic. *Journal of Logic and Computation*, 11(5):717–736, 2001.

[8] C. Areces, R. Gennari, J. Heguiabehere, and M. de Rijke. A simple ordering for deciding modal logic. Forthcoming.

[9] C. Areces, R. Gennari, J. Heguiabehere, and M. de Rijke. Tree-based heuristic in modal theorem proving. In W. Horn, editor, *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000)*, pages 199–203. IOS Press, 2000.

[10] Y. Auffray, P. Enjalbert, and J.-J. Hebrard. Strategies for modal resolution: Results and problems. *Journal of Automated Reasoning*, 6:1–38, 1990.

[11] F. Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. Technical Report RR-90-13, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Kaiserslautern, Germany, 1990. An abridged version appeared in *Proceedings of IJCAI-91*, pp. 446–451.

[12] F. Baader, J. Hladik, C. Lutz, and F. Wolter. From tableaux to automata for description logics. *Fundamenta Informaticae*, 57(2-4):247–279, 2003.

[13] F. Baader and S. Tobies. The inverse method implements the automata approach for modal satisfiability. In *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR-01)*, volume 2083 of *Lecture Notes in Artificial Intelligence*, pages 92–106. Springer, 2001.

[14] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.

[15] L. Bachmair and H. Ganzinger. Ordered chaining calculi for first-order theories of transitive relations. *Journal of the ACM*, 45(6):1007–1049, 1998.

[16] L. Bachmair and H. Ganzinger. Resolution theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 2, pages 19–99. Elsevier, 2001.

[17] L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder. Basic paramodulation. *Information and Computation*, 121(2):172–192, 1995.

[18] L. Bachmair, H. Ganzinger, and U. Waldmann. Superposition with simplification as a decision procedure for the monadic class with equality. In G. Gottlob, A. Leitsch, and D. Mundici, editors, *Proceedings of the Third Kurt Gödel Colloquium (KGC '93)*, volume 713 of *Lecture Notes in Computer Science*, pages 83–96. Springer, 1993.

[19] A. B. Baker. *Intelligent Backtracking on Constraint Satisfaction Problems: Experimental and Theoretical Results*. PhD thesis, University of Oregon, 1995.

[20] P. Balbiani, L. Farinas del Cerro, and A. Herzig. Declarative semantics for modal logic programs. In *Proceedings of the 1988 International Conference on Fifth Generation Computer Systems*, pages 507–514. ICOT, 1988.

[21] P. Balbiani, A. Herzig, and M. Lima Marques. TIM: The Toulouse Inference Machine for non-classical logic programming. In H. Boley and M. M. Richter, editors, *Proceedings of the International Workshop on Processing Declarative Knowledge (PDK '91)*, volume 567 of *Lecture Notes in Artificial Intelligence*, pages 366–382. Springer, 1991.

[22] M. Baldoni. *Normal Multimodal Logics: Automatic Deduction and Logic Programming Extension*. PhD thesis, Dipartimento di Informatica, Universit degli Studi di Torino, Torino, Italy, April 1998.

[23] M. Baldoni, L. Giordano, and A. Martelli. A framework for a modal logic programming. In M. J. Maher, editor, *Joint International Conference and Symposium on Logic Programming*, pages 52–66. MIT Press, 1996.

[24] M. Baldoni, A. Martelli, V. Patti, and L. Giordano. Programming rational agents in a modal action logic. *Annals of Mathematics and Artificial Intelligence*, 41(2-4):207–257, 2004.

[25] P. Balsiger, A. Heuerding, and S. Schwendimann. Logics Workbench 1.0. In H. de Swart, editor, *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX '98)*, volume 1397 of *Lecture Notes in Artificial Intelligence*, pages 35–35. Springer, 1998.

[26] P. Baumgartner, J. D. Horton, and B. Spencer. Merge path improvements for minimal model hyper tableaux. In N. V. Murray, editor, *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX '99)*, volume 1617 of *Lecture Notes in Artificial Intelligence*, pages 51–65. Springer, 1999.

[27] P. Baumgartner and R. A. Schmidt. Improved bottom-up model generation. In U. Furbach and N. Shankar, editors, *Proceedings of the Third International Joint Conference on Automated Reasoning (IJCAR 2006)*, Lecture Notes in Artificial Intelligence. Springer, 2006. To appear.

[28] A. Bouajjani, J.-C. Fernandez, and N. Halbwachs. Minimal model generation. In E. M. Clarke and R. P. Kurshan, editors, *Proceedings of the 2nd International Workshop on Computer Aided Verification (CAV '90)*, volume 531 of *Lecture Notes in Computer Science*, pages 197–203. Springer, 1991.

[29] T. Boy de la Tour. An optimality result for clause form translation. *Journal of Symbolic Computation*, 14:283–301, 1992.

[30] T. Braüner. A cut-free Gentzen formulation of the modal logic S5. *Logic Journal of the IGPL*, 8(5):629–643, 2000.

[31] F. Bry and A. Yahya. Positive unit hyperresolution tableaux for minimal model generation. *Journal of Automated Reasoning*, 25(1):35–82, 2000.

[32] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Daniele Nardi, and Riccardo Rosati. Description logic framework for information integration. In A. G. Cohn, L. K. Schubert, and S. C. Shapiro, editors, *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR '98)*, pages 2–13. Morgan Kaufmann, 1998.

[33] M. Cialdea. Resolution for some first order modal systems. *Theoretical Computer Science*, 85:213–229, 1991.

[34] E. M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

[35] E. M. Clarke and B.-H. Schlingloff. Model checking. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 24, pages 1635–1790. Elsevier, 2001.

[36] W. Conradie, V. Goranko, and D. Vakarelov. Elementary canonical formulae: a survey on syntactic, algorithmic, and model-theoretic aspects. In R. A. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing, editors, *Advances in Modal Logic, Volume 5*, pages 17–51. King's College Publications, London, 2005.

[37] W. Conradie, V. Goranko, and D. Vakarelov. Algorithmic correspondence and completeness in modal logic. i. the core algorithm SQEMA. *Logical Methods in Computer Science*, 2(1:5):1–26, 2006.

[38] S. A. Cook and R. A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979.

[39] H. B. Curry. The elimination theorem when modality is present. *Journal of Symbolic Logic*, 17:249–265, 1952.

[40] M. D'Agostino. Are tableaux an improvement on truth-tables? *Journal of Logic, Language, and Information*, 1:235–252, 1992.

[41] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem proving. *Communications of the ACM*, 5:394–397, 1962.

[42] M. Davis and H. Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7:201–215, 1960.

[43] G. De Giacomo. Eliminating 'converse' from converse PDL. *Journal of Logic, Language and Information*, 5(2):193–208, 1996.

[44] G. De Giacomo and M. Lenzerini. Boosting the correspondence between description logics and propositional dynamic logics (extended abstract). In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*, pages 205–212. AAAI Press, 1994.

[45] G. De Giacomo and F. Massacci. Combining deduction and model checking into tableaux and algorithms for converse-pdl. *Information and Computation*, 160(1–2):117–137, 2000.

[46] H. De Nivelle. *Ordering refinements of resolution*. PhD thesis, Technische Universiteit Delft, The Netherlands, 1996.

[47] H. De Nivelle. Splitting through new proposition symbols. In R. Nieuwenhuis and A. Voronkov, editors, *Proceedings of the 8th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2001)*, volume 2250 of *Lecture Notes in Artificial Intelligence*, pages 172–185. Springer, 2001.

[48] H. De Nivelle and M. de Rijke. Deciding the guarded fragment by resolution. *Journal of Symbolic Computation*, 35(1):21–58, 2003.

[49] H. De Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. *Logic Journal of the IGPL*, 8(3):265–292, 2000.

[50] F. Debart, P. Enjalbert, and M. Lescot. Multimodal logic programming using equational and order-sorted logic. *Theoretical Computer Science*, 105(1):141–166, 1992.

[51] A. Degtyarev and A. Voronkov. The inverse method. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 21, pages 179–272. Elsevier Science, 2001.

[52] N. Dershowitz and D. A. Plaisted. Rewriting. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 9, pages 533–607. Elsevier, 2001.

[53] J. Dix, S. Kraus, and V. S. Subrahmanian. Temporal Agent Reasoning. *Artificial Intelligence*, 127(1):87–135, 2001.

[54] C. Dixon, M. Fisher, and M. Wooldridge. Resolution for temporal logics of knowledge. *Journal of Logic and Computaton*, 8(3):345–372, 1998.

[55] P. Doherty, W. Lukaszewics, and A. Szalas. Computing circumscription revisited: A reduction algorithm. *Journal of Automated Reasoning*, 18(3):297–336, 1997.

[56] F. M. Donini and F. Massacci. Exptime tableaux for $\mathcal{ALC}$. *Artificial Intelligence*, 124(1):87–138, 2000.

[57] A. Dovier, C. Piazza, and A. Policriti. An efficient algorithm for computing bisimulation equivalence. *Theoretical Computer Science*, 311(1-3):221–256, 2004.

[58] T. Engel. Quantifier elimination in second-order predicate logic. Diplomarbeit, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, 1996.

[59] P. Enjalbert and L. Fariñas del Cerro. Modal resolution in clausal form. *Theoretical Computer Science*, 65(1):1–33, 1989.

[60] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1996.

[61] L. Fariñas del Cerro. A simple deduction method for modal logic. *Information Processing Letters*, 14(2):49–51, April 1982.

[62] L. Fariñas del Cerro. MOLOG: A system that extends PROLOG with modal logic. *New Generation Computing*, 4:35–50, 1986.

[63] L. Fariñas del Cerro and A. Herzig. MOLOG. `http://www.irit.fr/ACTIVITES/EQ_ALG/Herzig/molog.html`.

[64] L. Fariñas del Cerro and A. Herzig. Modal deduction with applications in epistemic and temporal logics. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming: Epistemic and Temporal Reasoning*, volume 4, pages 499–594. Clarendon Press, Oxford, 1995.

[65] C. Fermüller, A. Leitsch, U. Hustadt, and T. Tammet. Resolution decision procedures. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 25, pages 1791–1849. Elsevier, 2001.

[66] J. Fernandez and L. Mounier. Verification bisimulations 'on the fly'. In J. Quemada, J. A. Mañas, and E. Vázquez, editors, *In Proceedings of the Third International Conference on Formal Description Techniques (FORTE '90)*, pages 95–110. North-Holland, 1990.

[67] J.-C. Fernandez. An implementation of an efficient algorithm for bisimulation equivalence. *Science of Computer Programming*, 13(2–3):219–236, 1990.

[68] M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Science*, 18:194–211, 1979.

[69] M. Fisher, C. Ghidini, and B. Hirsch. Organising logic-based agents. In M. G. Hinchey, J. L. Rash, W.t Truszkowski, C. Rouff, and D. F. Gordon-Spears, editors, *Revised Papers of the Second International Workshop on Formal Approaches to Agent-Based Systems (FAABS 2002)*, volume 2699 of *Lecture Notes in Artificial Intelligence*, pages 15–27. Springer, 2003.

[70] M. Fisher and R. Owens. An introduction to executable modal and temporal logics. In *Proceedings of the IJCAI-93 Workshop on Executable Modal and Temporal Logics*, volume 897 of *Lecture Notes in Artificial Intelligence*, pages 1–20. Springer, 1995.

[71] K. Fisler and M. Y. Vardi. Bisimulation and model checking. In L. Pierre and T. Kropf, editors, *Proceedings of the 10th IFIP WG 10.5 Advanced Research Working Conference on Correct Hardware Design and Verification Methods (CHARME '99)*, volume 1703 of *Lecture Notes in Computer Science*, pages 338–341. Springer, 1999.

[72] M. Fitting. Destructive modal resolution. *Journal of Logic and Computation*, 1(1):83–97, 1990.

[73] E. Franconi and G. Ng. The i.com tool for intelligent conceptual modelling. In *Working Notes of the ECAI2000 Workshop on Knowledge Representation Meets Databases (KRDB2000)*, pages 45–53. CEUR, 2000.

[74] J. W. Freeman. *Improvements to Propositional Satisfiability Search Algorithms*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, 1995.

[75] D. M. Gabbay and H. J. Ohlbach. Quantifier elimination in second-order predicate logic. *South African Computer Journal*, 7:35–43, 1992. Also published in B. Nebel, C. Rich, W. R. Swartout, editors, *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR '92)*, pages 425–436. Morgan Kaufmann, 1992.

[76] H. Ganzinger and H. De Nivelle. A superposition decision procedure for the guarded fragment with equality. In *Proceedings of the Fourteenth Annual IEEE Symposium on Logic in Computer Science (LICS '99)*, pages 295–303. IEEE Computer Society Press, 1999.

[77] H. Ganzinger, U. Hustadt, C. Meyer, and R. A. Schmidt. A resolution-based decision procedure for extensions of K4. In M. Zakharyaschev, K. Segerberg, M. de Rijke, and H. Wansing, editors, *Advances in Modal Logic, Volume 2*, volume 119 of *Lecture Notes*, chapter 9, pages 225–246. CSLI Publications, Stanford, 2001.

[78] J. Gaschnig. Performance measurement and analysis of certain search algorithms. Technical report CMU-CS-79-124, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA, 1979.

[79] C. Geissler and K. Konolige. A resolution method for quantified modal logics of knowledge and belief. In J. Y. Halpern, editor, *Proceedings of the 1st Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 309–324. Morgan Kaufmann, 1986.

[80] G. Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935. English translation in M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131, North-Holland, 1969.

[81] L. Georgieva, U. Hustadt, and R. A. Schmidt. Computational space efficiency and minimal model generation for guarded formulae. In R. Nieuwenhuis and A. Voronkov, editors, *Proceedings of the 8th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2001)*, volume 2250 of *Lecture Notes in Artificial Intelligence*, pages 85–99. Springer, 2001.

[82] L. Georgieva, U. Hustadt, and R. A. Schmidt. A new clausal class decidable by hyperresolution. In A. Voronkov, editor, *Proceedings of the 18th International Conference on Automated Deduction (CADE-18)*, volume 2392 of *Lecture Notes in Artificial Intelligence*, pages 260–274. Springer, 2002. The long version is Preprint CSPP-18, University of Manchester, UK.

[83] L. Georgieva, U. Hustadt, and R. A. Schmidt. Hyperresolution for guarded formulae. *Journal of Symbolic Computation*, 36(1–2):163–192, 2003.

[84] M. Gergatsoulis. Temporal and modal logic programming languages. In A. Kent and J. G. Williams, editors, *Encyclopedia of Microcomputers*, chapter Volume 27, Supplement 6, pages 393–408. Marcel Dekker, 2001.

[85] E. Giunchiglia, F. Giunchiglia, R. Sebastiani, and A. Tacchella. Sat vs. translation based decision procedures for modal logics: A comparative evaluation. *Journal of Applied Non-Classical Logics*, 10(2):145–172, 2000.

[86] E. Giunchiglia and A. Tacchella. A subset-matching size-bounded cache for satisfiability in modal logics. In R. Dyckhoff, editor, *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2000)*, volume 1847 of *Lecture Notes in Artificial Intelligence*, pages 237–251. Springer, 2000.

[87] F. Giunchiglia and R. Sebastiani. Building decision procedures for modal logics from propositional decision procedures - the case study of modal K. In M. McRobbie and J. K. Slaney, editors, *Proceedings of the 13th Conference on Automated Deduction (CADE-13)*, volume 1104 of *Lecture Notes in Artificial Intelligence*, pages 583–597, New Brunswick, NJ, USA, 1996. Springer.

[88] V. Goranko, U. Hustadt, R. Schmidt, and D. Vakarelov. SCAN is complete for all Sahlqvist formulae. In R. Berghammer, B. Möller, and G. Struth, editors, *Revised Selected Papers of the 7th International Seminar on Relational Methods in Computer Science and the 2nd International Workshop on Kleene Algebra*, volume 3051 of *Lecture Notes in Computer Science*, pages 149–162. Springer, 2004.

[89] R. Goré. Tableau methods for modal and temporal logics. In M. D'Agostino, D. Gabbay, R. Hähnle, and J. Posegga, editors, *Handbook of Tableau Methods*, pages 297–396. Kluwer, 1999.

[90] V. Haarslev and R. Möller. Consistency testing: The RACE experience. In R. Dyckhoff, editor, *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2000)*, volume 1847 of *Lecture Notes in Artificial Intelligence*, pages 1–18. Springer, 2000.

[91] Volker Haarslev and Ralf Möller. High performance reasoning with very large knowledge bases: A practical case study. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*, pages 161–168. Morgan Kaufmann, 2001.

[92] R. Hähnle. Tableaux and related methods. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, chapter 3, pages 101–178. Elsevier Science Publishers (North-Holland), Amsterdam, 2001.

[93] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logic of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.

[94] R. Hasegawa, H. Fujita, and M. Koshimura. Efficient minimal model generation using branching lemmas. In D. McAllester, editor, *Proceedings of the 17th International Conference on Automated Deduction (CADE-17)*, volume 1831 of *Lecture Notes in Artificial Intelligence*, pages 184–199. Springer, 2000.

[95] A. Herzig. A new decidable fragment of first order logic, June 1990. In Abstracts of the 3rd Logical Biennial, Summer School & Conference in honour of S. C. Kleene, Varna, Bulgaria.

[96] A. Heuerding. *Sequent Calculi for Proof Search in Some Modal Logics*. PhD thesis, Universität Bern, Switzerland, 1996.

[97] A. Heuerding, G. Jäger, S. Schwendimann, and M. Seyfried. The Logics Workbench LWB: A snapshot. *Euromath Bulletin*, 2(1):177–186, 1996.

[98] A. Heuerding, M. Seyfried, and H. Zimmermann. Efficient loop-check for backward proof search in some non-classical propositional logics. In P. Miglioli, U. Moscato, D. Mundici, and M. Ornaghi, editors, *Proceedings of the 5th International Workshop on Theorem Proving with Analytic Tableaux and Related Methods (TABLEAUX '96)*, volume 1071 of *Lecture Notes in Artificial Intelligence*, pages 210–225. Springer, 1996.

[99] J. Hladik. Implementation and optimisation of a tableau algorithm for the guarded fragment. In U. Egly and C. G. Fermüller, editors, *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2002)*, volume 2381 of *Lecture Notes in Artificial Intelligence*, pages 145–159. Springer, 2002.

[100] Jörg Hoffmann and Jana Koehler. A new method to index and query sets. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 462–467. Morgan Kaufmann, 1999.

[101] B. Hollunder and F. Baader. Qualifying number restrictions in concept languages. In *Proceedings of the 2nd International Conference on the Principles of Knowledge Representation and Reasoning (KR '91)*, pages 335–346. Morgan Kaufmann, 1991.

[102] I. Horrocks. *Optimising Tableaux Decision Procedures for Description Logics*. PhD thesis, University of Manchester, 1997.

[103] I. Horrocks. Implementation and optimisation techniques. In Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 306–346. Cambridge University Press, 2003.

[104] I. Horrocks and P. F. Patel-Schneider. Optimizing description logic subsumption. *Journal of Logic and Computation*, 9(3):267–293, 1999.

[105] I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. *Journal of Logic and Computation*, 9(3), 1999.

[106] I. Horrocks and U. Sattler. Ontology reasoning in the $\mathcal{SHOQ}(D)$ description logic. In B. Nebel, editor, *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*, pages 199–204. Morgan Kaufmann, 2001.

[107] I. Horrocks and U. Sattler. Optimised reasoning for $\mathcal{SHIQ}$. In F. van Harmelen, editor, *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI 2002)*, pages 277–281. IOS Press, July 2002.

[108] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, *Proceedings of the 6th Int. Conf. on Logic for Programming and Automated Reasoning (LPAR '99)*, volume 1705 of *Lecture Notes in Artificial Intelligence*, pages 161–180. Springer, 1999.

[109] I. Horrocks and S. Tobies. Reasoning with axioms: Theory and practice. In A. G. Cohn, F. Giunchiglia, and B. Selman, editors, *Proceedings of the 7th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2000)*, pages 285–296. Morgan Kaufmann, 2000.

[110] Ian Horrocks and Ulrike Sattler. A tableaux decision procedure for $\mathcal{SHOIQ}$. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*, pages 448–453, 2005.

[111] U. Hustadt. *Resolution-Based Decision Procedures for Subclasses of First-Order Logic*. PhD thesis, Univ. d. Saarlandes, Saarbrücken, Germany, 1999.

[112] U. Hustadt, B. Motik, and U. Sattler. Reducing $\mathcal{SHIQ}^-$ description logic to disjunctive datalog programs. In D. Dubois, C. Welty, and M.-A. Williams, editors, *Proceedings of the 9th International Conference on Knowledge Representation and Reasoning (KR 2004)*, pages 152–162. AAAI Press, 2004.

[113] U. Hustadt, B. Motik, and U. Sattler. A decomposition rule for decision procedures by resolution-based calculi. In F. Baader and A. Voronkov, editors, *Proceedings of the 11th International Conference on Logic for Programming, Artificial Intelligence (LPAR 2004)*, volume 3452 of *Lecture Notes in Artificial Intelligence*, pages 21–35. Springer, 2005.

[114] U. Hustadt and R. A. Schmidt. On evaluating decision procedures for modal logic. In M. E. Pollack, editor, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 202–207. Morgan Kaufmann, 1997.

[115] U. Hustadt and R. A. Schmidt. Simplification and backjumping in modal tableau. In H. de Swart, editor, *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX '98)*, volume 1397 of *Lecture Notes in Artificial Intelligence*, pages 187–201. Springer, 1998.

[116] U. Hustadt and R. A. Schmidt. An empirical analysis of modal theorem provers. *Journal of Applied Non-Classical Logics*, 9(4):479–522, 1999.

[117] U. Hustadt and R. A. Schmidt. Maslov's class K revisited. In H. Ganzinger, editor, *Proceedings of the 16th International Conference on Automated Deduction (CADE-16)*, volume 1632 of *Lecture Notes in Artificial Intelligence*, pages 172–186. Springer, 1999.

[118] U. Hustadt and R. A. Schmidt. On the relation of resolution and tableaux proof systems for description logics. In T. Dean, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 110–115. Morgan Kaufmann, 1999.

[119] U. Hustadt and R. A. Schmidt. Issues of decidability for description logics in the framework of resolution. In R. Caferra and G. Salzer, editors, *Automated Deduction in Classical and Non-Classical Logics*, volume 1761 of *Lecture Notes in Artificial Intelligence*, pages 191–205. Springer, 2000.

[120] U. Hustadt and R. A. Schmidt. MSPASS: Modal reasoning by translation and first-order resolution. In R. Dyckhoff, editor, *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2000)*, volume 1847 of *Lecture Notes in Artificial Intelligence*, pages 67–71. Springer, 2000.

[121] U. Hustadt and R. A. Schmidt. Using resolution for testing modal satisfiability and building models. *Journal of Automated Reasoning*, 28(2):205–232, 2002.

[122] U. Hustadt, R. A. Schmidt, and L. Georgieva. A survey of decidable first-order fragments and description logics. *Journal of Relational Methods in Computer Science*, 1:251–276, 2004.

[123] W. H. Joyner Jr. Resolution strategies as decision procedures. *Journal of the ACM*, 23(3):398–417, 1976.

[124] Y. Kazakov and H. De Nivelle. A resolution decision procedure for the guarded fragment with transitive guards. In D. A. Basin and M. Rusinowitch, editors, *Proceedings of the Second International Joint Conference on Automated Reasoning (IJCAR 2004)*, volume 3097 of *Lecture Notes in Artificial Intelligence*, pages 122–136. Springer, 2004.

[125] Holger Knublauch, Ray Fergerson, Natalya Noy, and Mark Musen. The protégé OWL plugin: An open development environment for semantic web applications. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *Proceedings of the 2004 International Semantic Web Conference (ISWC 2004)*, volume 3298 of *Lecture Notes in Computer Science*, pages 229–243. Springer, 2004.

[126] K. Konolige. Resolution and quantified epistemic logics. In J. H. Siekmann, editor, *Proceedings of the 8th International Conference on Automated Deduction (CADE-8)*, volume 230 of *Lecture Notes in Computer Science*, pages 199–208. Springer, 1986.

[127] M. Kracht. *Tools and Techniques in Modal Logic*, volume 142 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1999.

[128] O. Kupferman and M.Y. Vardi. Weak alternating automata and tree automata emptiness. In *Proceedings of the 30th ACM SIGACT Symposium on Theory of Computing (STOC-98)*, pages 224–233. ACM Press, 1998.

[129] R. E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal of Computing*, 6(3):467–480, 1977.

[130] D. Lee and M. Yannakakis. Online minimization of transition systems (extended abstract). In *Proceedings of the 24th Annual ACM symposium on Theory of Computing*, pages 264–274. ACM Press, 1992.

[131] Thorsten Liebig and Olaf Noppens. Ontotrack: Combining browsing and editing with reasoning and explaining for OWL Lite ontologies. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *Proceedings of the 2004 International Semantic Web Conference (ISWC 2004)*, volume 3298 of *Lecture Notes in Computer Science*, pages 244–258. Springer, 2004.

[132] C. Lutz. Complexity of terminological reasoning revisited. In H. Ganzinger, D. A. McAllester, and A. Voronkov, editors, *Proceedings of the 6th Int. Conf. on Logic for Programming and Automated Reasoning (LPAR '99)*, volume 1705 of *Lecture Notes in Artificial Intelligence*, pages 181–200. Springer, 1999.

[133] C. Lutz. NExpTime-complete description logics with concrete domains. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR-01)*, volume 2083 of *Lecture Notes in Artificial Intelligence*, pages 45–60. Springer, 2001.

[134] C. Lutz, U. Sattler, and S. Tobies. A suggestion of an n-ary description logic. In P. Lambrix, A. Borgida, M. Lenzerini, R. Möller, and P. Patel-Schneider, editors, *Proceedings of the 1999 International Workshop on Description Logics (DL '99)*, pages 81–85. Linköping University, 1999.

[135] S. Ju. Maslov. The inverse method for establishing deducibility for logical calculi. In V. P. Orevkov, editor, *The Calculi of Symbolic Logic I: Proc. of the Steklov Institute of Mathematics edited by I. G. Petrovskiĭ and S. M. Nikol'skiĭ, Nr. 98 (1968)*, pages 25–96. Amer. Math. Soc., Providence, Rhode Island, 1971.

[136] W. McCune and O. Shumsky. Ivy: A preprocessor and proof checker for first-order logic. In M. Kaufmann, P. Manolios, and J. Moore, editors, *Computer-Aided Reasoning: ACL2 Case Studies*, pages 265–282. Kluwer, 2000.

[137] Deborah L. McGuinness and Jon R. Wright. An industrial strength description logic-based configuration platform. *IEEE Intelligent Systems*, pages 69–77, 1998.

[138] P. Mika, D. Oberle, A. Gangemi, and M. Sabou. Foundations for service ontologies: Aligning OWL-S to dolce. In S.I. Feldman, M. Uretsky, M. Najork, and C.E. Wills, editors, *Proceedings of the Thirteenth International World Wide Web Conference (WWW 2004)*, pages 563–572. ACM Press, 2004.

[139] G. Mints. Gentzen-type systems and resolution rules. Part I: Propositional lo gic. In *Proceedings of COLOG-88*, volume 417 of *Lecture Notes in Computer Science*, pages 198–231. Springer, 1990.

[140] G. Mints, V. Orevkov, and T. Tammet. Transfer of sequent calculus strategies to resolution for S4. In H. Wansing, editor, *Proof Theory of Modal Logic*, volume 2 of *Applied Logic Series*, pages 17–31. Kluwer, 1996.

[141] L. Nguyen. MProlog. `http://www.mimuw.edu.pl/~nguyen/mprolog/`.

[142] L. Nguyen. A fixpoint semantics and an sld-resolution calculus for modal logic programs. *Fundamenta Informaticae*, 55(1):63–100, 2003.

[143] L. A. Nguyen. The modal logic programming system MProlog. In J. J. Alferes and J. A. Leite, editors, *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA 2004)*, volume 3229 of *Lecture Notes in Artificial Intelligence*, pages 266–278. Springer, 2004.

[144] I. Niemelä. A tableau calculus for minimal model reasoning. In *Proceedings of the 5th International Workshop on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX '96)*, volume 1071 of *Lecture Notes in Artificial Intelligence*, pages 278–294. Springer, 1996.

[145] A. Nonnengart. First-order modal logic theorem proving and functional simulation. In R. Bajcsy, editor, *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 80–85. Morgan Kaufmann, 1993.

[146] A. Nonnengart. How to use modalities and sorts in prolog. In C. MacNish, D. Pearce, and L. M. Pereira, editors, *Proceedings of the European Workshop on Logics in Artificial Intelligence (JELIA '94)*, volume 838 of *Lecture Notes in Artificial Intelligence*, pages 365–378. Springer, 1994.

[147] A. Nonnengart, H. J. Ohlbach, and A. Szalas. Quantifier elimination for second-order predicate logic. To appear in *Logic, Language and Reasoning: Essays in honour of Dov Gabbay*, Part I, Kluwer.

[148] H. J. Ohlbach, A. Nonnengart, M. de Rijke, and D. Gabbay. Encoding two-valued nonclassical logics in classical logic. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 21, pages 1403–1486. Elsevier Science, 2001.

[149] H. J. Ohlbach and R. A. Schmidt. Functional translation and second-order frame properties of modal logics. *Journal of Logic and Computation*, 7(5):581–603, 1997.

[150] H. J. Ohlbach, R. A. Schmidt, and U. Hustadt. Translating graded modalities into predicate logic. In H. Wansing, editor, *Proof Theory of Modal Logic*, volume 2 of *Applied Logic Series*, pages 253–291. Kluwer, 1996.

[151] M. Ohnishi and K. Matsumoto. Gentzen method in modal calculi. *Osaka Mathematical Journal*, 9:113–130, 1957.

[152] M. Ohnishi and K. Matsumoto. Gentzen method in modal calculi II. *Osaka Mathematical Journal*, 11:115–120, 1959.

[153] F. Oppacher and E. Suen. HARP: A tableau-based theorem prover. *Journal of Automated Reasoning*, 4:69–100, 1988.

[154] M. A. Orgun and W. Ma. An overview of temporal and modal logic programming. In D. M. Gabbay and H. J. Ohlbach, editors, *Proceedings of the First International Conference on Temporal Logic (ICTL '94)*, volume 827 of *Lecture Notes in Artificial Intelligence*, pages 445–479. Springer, 1994.

[155] S. Owre, S. Rajan, J. M. Rushby, N. Shankar, and M. K. Srivas. PVS: Combining specification, proof checking, and model checking. In R. Alur and T. A. Henzinger, editors, *Proceedings of the 8th International Conference on Computer-Aided Verification (CAV '96)*, volume 1102 of *Lecture Notes in Computer Science*, pages 411–414. Springer, 1996.

[156] R. Paige and R. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–989, 1987.

[157] R. Paige, R. E. Tarjan, and R. Bonic. A linear time solution to the single function coarsest partition problem. *Theoretical Computer Science*, 40:67–84, 1985.

[158] G. Pan and M. Y. Vardi. Optimizing a BDD-based modal solver. In F. Baader, editor, *Proceedings of the 19th International Conference on Automated Deduction (CADE-19)*, volume 2741 of *Lecture Notes in Artificial Intelligence*, pages 75–89. Springer, 2003.

[159] P. F. Patel-Schneider and I. Horrocks. DLP and FaCT. In *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX '99)*, volume 1397 of *Lecture Notes in Artificial Intelligence*, pages 19–23. Springer, 1999.

[160] Pellet OWL reasoner. Maryland Information and Network Dynamics Lab, 2003.

[161] D. A. Plaisted and S. Greenbaum. A structure-preserving clause form translation. *Journal of Symbolic Computation*, 2:293–304, 1986.

[162] V. R. Pratt. Models of program logics. In *Proceedings of the 20th Annual Symposium on Foundations of Computer Science*, 1979.

[163] A. Prior. *Past, Present and Future*. Oxford University Press, 1967.

[164] P. Prosser. Hybrid algorithms for the constraint satisfaction problem. *Computational Intelligence*, 9(3):268–299, 1993.

[165] W. V. Quine. Variables explained away. In *Proceedings of the American Philosophy Society*, volume 104, pages 343–347, 1960.

[166] W. V. Quine. Algebraic logic and predicate functors. In R. Rudner and I. Scheffler, editors, *Logic and Art: Esssays in Honor of Nelson Goodman*. Bobbs-Merrill, 1971.

[167] A. L. Rector, W. A. Nowlan, and A. Glowinski. Goals for concept representation in the GALEN project. In *Proceedings of the 17th Annual Symposium on Computer Applications in Medical Care (SCAMC '93)*, pages 414–418, 1993.

[168] A. Riazanov and A. Voronkov. Splitting without backtracking. In B. Nebel, editor, *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*, pages 611–617. Morgan Kaufmann, 2001.

[169] A. Riazanov and A. Voronkov. The design and implementation of Vampire. *AI Communications*, 15(2–3):91–110, 2002.

[170] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12:23–41, 1965.

[171] U. Sattler. Description logics for the representation of aggregated objects. In W. Horn, editor, *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000)*, pages 239–243. IOS Press, 2000.

[172] W. J. Savitch. Relationsship between nondeterministic and deterministic tape complexities. *Journal of Computer and System Science*, 4:177–192, 1970.

[173] K. Schild. A correspondence theory for terminological logics: Preliminary report. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 466–471. Morgan Kaufmann, 1991.

[174] R. A. Schmidt. MSPASS. `http://www.cs.man.ac.uk/~schmidt/mspass`.

[175] R. A. Schmidt. *Optimised Modal Translation and Resolution*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1997.

[176] R. A. Schmidt. E-unification for subsystems of S4. In T. Nipkow, editor, *Proceedings of the 9th International Conference on Rewriting Techniques and Applications (RTA '98)*, volume 1379 of *Lecture Notes in Computer Science*, pages 106–120. Springer, 1998.

[177] R. A. Schmidt. Decidability by resolution for propositional modal logics. *Journal of Automated Reasoning*, 22(4):379–396, 1999.

[178] R. A. Schmidt. Decidability by resolution for propositional modal logics. *Journal of Automated Reasoning*, 22(4):379–396, 1999.

[179] R. A. Schmidt and U. Hustadt. A resolution decision procedure for fluted logic. In D. McAllester, editor, *Proceedings of the 17th International Conference on Automated Deduction (CADE-17)*, volume 1831 of *Lecture Notes in Artificial Intelligence*, pages 433–448. Springer, 2000.

[180] R. A. Schmidt and U. Hustadt. Mechanised reasoning and model generation for extended modal logics. In H. C. M. de Swart, E. Orlowska, G. Schmidt, and M. Roubens, editors, *Theory and Applications of Relational Structures as Knowledge Instruments*, volume 2929 of *Lecture Notes in Computer Science*, pages 38–67. Springer, 2003.

[181] R. A. Schmidt and U. Hustadt. A principle for incorporating axioms into the first-order translation of modal formulae. In F. Baader, editor, *Proceedings of the 19th International Conference on Automated Deduction (CADE-19)*, volume 2741 of *Lecture Notes in Artificial Intelligence*, pages 412–426. Springer, 2003. The long version is Preprint CSPP-22, University of Manchester, UK.

[182] R. A. Schmidt and U. Hustadt. First-order resolution methods for modal logics. In A. Podelski, A. Voronkov, and R. Wilhelm, editors, *Volume in memoriam of Harald Ganzinger*, Lecture Notes in Computer Science. Springer, 2006. To appear.

[183] S. Schulz. E: A Brainiac theorem prover. *Journal of AI Communications*, 15(2/3):111–126, 2002.

[184] E. Spaan. *Complexity of Modal Logics*. PhD thesis, University of Amsterdam, 1993.

[185] A. Szalas. On the correspondence between modal and classical logic: An automated approach. *Journal of Logic and Computation*, 3(6):605–620, 1993.

[186] S. Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen, 2001.

[187] J. van Benthem. Temporal logic. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 4*, pages 241–350. Oxford Scientific Publishers, 1996.

[188] M. Y. Vardi. Reasoning about the past with two-way automata. In *Proceedings of the 25th Int. Colloq. on Automata, Languages, and Programming*, volume 1443 of *Lecture Notes in Computer Science*, pages 628–641. Springer, 1998.

[189] U. Visser, H. Stuckenschmidt, G. Schuster, and T. Vögele. Ontologies for geographic information processing. *Computers & Geosciences*, 28(1):103–117, 2002.

[190] A. Voronkov. Theorem proving in non-standard logics based on the inverse method. In *Proceedings of the 11th Conference on Automated Deduction (CADE-11)*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 648–662. Springer, 1992.

[191] D. H. D. Warren. An abstract prolog instruction set. Technical Note 309, SRI International, Menlo Park, CA, USA, 1983.

[192] C. Weidenbach. SPASS. `http://spass.mpi-sb.mpg.de`.

[193] C. Weidenbach. Combining superposition, sorts and splitting. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 27, pages 1965–2013. Elsevier, 2001.

[194] C. Weidenbach et al. System description: SPASS version 1.0.0. In H. Ganzinger, editor, *Proceedings of the 16th International Conference on Automated Deduction (CADE-16)*, volume 1632 of *Lecture Notes in Artificial Intelligence*, pages 378–382. Springer, 1999.

[195] W. Wong. Validation of HOL proofs by proof checking. *Formal Methods in System Design*, 14(2):193–212, 1999.

[196] M. Wooldridge. *Reasoning about rational agents*. MIT Press, 2000.

[197] C. Wroe, C. A. Goble, A. Roberts, and M. Greenwood. A suite of DAML+OIL ontologies to describe bioinformatics web services and data. *International Journal of Cooperative Information Systems*, 12(2):597–624, 2003.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

5

MODEL THEORY OF MODAL LOGIC

Valentin Goranko and Martin Otto

1	Semantics of modal logic	252
1.1	Modal languages	252
1.2	Kripke frames and structures	252
1.3	The standard translations into first- and second-order logic	253
1.4	Theories, equivalence and definability	254
1.5	Polyadic modalities	256
2	Bisimulation and basic model constructions	256
2.1	Bisimulation and invariance	257
2.2	Classical truth-preserving constructions	258
2.3	Proving non-definability	262
3	Bisimulation: a closer look	263
3.1	Bisimulation games	263
3.2	Finite bisimulations and characteristic formulae	264
3.3	Finite model property	266
3.4	Finite versus full bisimulation	270
3.5	Largest bisimulations as greatest fixed points	273
3.6	Bisimulation quotients and canonical representatives	274
3.7	Robinson consistency, local interpolation, and Beth definability	276
3.8	Bisimulation-safe modal operators	278
4	Modal logic as a fragment of first-order logic	278
4.1	Finite variable fragments of first-order logic	278
4.2	The van Benthem–Rosen characterisation theorem	281
4.3	Guarded fragments of first-order logic	285
5	Variations, extensions, and comparisons of modal logics	288
5.1	Variations through refined notions of bisimulation	289
5.2	Extensions beyond first-order	290
5.3	Model theoretic criteria	295
6	Further model-theoretic constructions	296
6.1	Ultrafilter extensions	296
6.2	Ultraproducts	298
6.3	Modal saturation and bisimulations	299
6.4	Modal definability of properties of Kripke structures	301
7	General frames	302
7.1	General frames as semantic structures in modal logic	303
7.2	Constructions and truth preservation results on general frames	304
7.3	Special types of general frames and persistence of modal formulae	306
8	Modal logic on frames	310
8.1	Modal definability of frame properties	310
8.2	Modal logic versus first-order logic on frames	311
8.3	Modal logic and first-order logic with least fixed points	316
8.4	Modal logic and second-order logic	317

9 Finite model theory of modal logics . . . 318
9.1 Finite versus classical model theory . . . 318
9.2 Capturing bisimulation invariant Ptime . . . 320
9.3 0-1 laws in modal logic . . . 321

INTRODUCTION

Model theory is about semantics; it studies the interplay between a logical language (logic) and the models (structures) for that language. Key issues therefore are *expressiveness* and *definability.* At the basic level these concern the questions which structural properties are expressible and which classes of structures are definable in the logic. These basic questions immediately lead to the study of *model constructions*; to the *analysis of models and of model classes* for given formulae or theories; to notions of *equivalence between structures* with respect to the truth of formulae; and to the study of preservation phenomena.

Modal logics[1] come as members of a loosely knit family and have various links to other logics – classical first- and second-order logic as well as, for instance, temporal and process logics stemming from particular applications. Correspondingly, the key issues mentioned above may also be studied comparatively, both within the family and in relation to other relevant logics. Such a comparative view can support an understanding of the internal coherence of the rich family of modal logics. It also offers a perspective to place modal logics in the wider logical and model theoretic context.

In regard to the coherence of the family of modal logics, it is important to understand in model theoretic terms what it is that makes a logic 'modal'. For that aim we devote a major part of this chapter to the discussion of *bisimulation.* Many other features of the 'modal character' can be understood in terms of bisimulation invariance; this is true most notably of the *local* and *restricted nature of quantification.* Due to these features modal logic enjoys very specific features, and in many respects its model theory can be developed along lines that have no direct counterparts in classical model theory.

In regard to the wider logical context, there is a rich body of classical work in modal model theory that measures modal logic against the backdrop of classical first- and second-order logic into which it can be naturally embedded. But, beside this 'classical picture', there are also many links with other logics, partly designed for other purposes or studied with a different perspective from that of classical model theory.

In the classical picture, both first- and second-order logic have their role to play. This is because modal logic actually offers several distinct semantic levels, as will be reviewed in the following section which provides an introduction to the model theoretic semantics of modal logic. So, a modal formula is traditionally viewed in four different ways, subject to two orthogonal dichotomies – *Kripke structures* (also called *Kripke models*) versus *Kripke frames* and *local* versus *global.*

The fundamental semantic notion in basic modal logic is *truth of a formula at a state in a Kripke structure*; this notion is *local* and of a *first-order* nature. Semantics in Kripke frames is obtained, if instead one looks at all possible propositional valuations

[1]In this chapter we use the term *modal logic* (despite the established tradition in the literature on modal logic) in a typical model-theoretic sense, as a (propositional) modal language equipped with suitable relational (Kripke) semantics, rather than proof systems over such languages, determined by a set of axioms and inference rules, such as K, S4, etc. We refer to the latter as 'axiomatic extensions'.

over the given frame (in effect an abstraction through implicit universal second-order quantification over all valuations); this semantics, accordingly, is of essentially *second-order* nature. On the other hand, the passage from local to *global* semantics is achieved if one looks at truth in all states (an abstraction through implicit universal first-order quantification over all states).

While all these semantic levels are ultimately based on the local semantics in Kripke structures, the two independent directions of generalisation, and in particular the divide between the (first-order) Kripke structure semantics and the (second-order) frame semantics, give rise to very distinct model theoretic flavours, each with their own tradition in the model theory of modal logic. Still, these two semantics meet through the notion of a *general frame* (closely related to a *modal algebra*).

History. The origins of model theory of modal logic go back to the fundamental papers of Jónsson and Tarski [78, 79], and Kripke [86, 87] laying the foundations of the *relational (Kripke) semantics*, followed by the classical work of Lemmon and Scott [91].

Some of the most influential themes and directions of the classical development of the model theory of modal logic in the 1970/80s have been: the *completeness theory* of modal axiomatic systems with respect to the frame-based semantics of modal logic, and the closely related *correspondence theory* between that semantics and first-order logic [117, 28, 123, 124, 113, 42, 51, 125, 127, 128]; and the *duality theory* between Kripke frames and modal algebras, via general frames [42, 43, 44, 45, 114]. Also at that time, the *theory of bisimulations and bisimulation invariance* emerged in the semantic analysis of modal languages in [125, 128]. For detailed historical and bibliographical notes see [5], and the survey [49] for a recent and comprehensive historical account of the development of modal logic, and in particular its model theory.

Overview. The sections of this chapter are roughly arranged in three parts or main tracks, reflecting the semantic distinctions outlined above.

The first part provides a common basic introduction to some of the key notions, in particular the different levels of semantics in section 1, followed by the concept of bisimulation and bisimulation respecting model constructions in section 2. This more general thread is taken up again in section 6 with some more advanced model constructions, and also in the final section 9 devoted to some ideas in the finite model theory of modal logic.

A second track, comprising sections 3 to 5, is primarily devoted to modal logic as a logic of Kripke structures (first-order semantics): section 3 continues the bisimulation theme; section 4 is specifically devoted to the role of modal logic as a fragment of first-order logic; section 5 illustrates some of the richness of modal logics over Kripke structures in terms of variations and extensions.

The third track is devoted to a study of modal logic as a logic of frames (the second-order semantics). This comprises more advanced constructions such as ultrafilter extensions and ultraproducts in section 6, basic model theory of general frames in section 7, and a survey of classical results on frame definability and relations with second-order logic in section 8.

Most of the other chapters in this handbook supplement this chapter with important model-theoretic topics and results. In particular, we refer the reader to Chapters 1, 3, 6, 7 and 8.

1 SEMANTICS OF MODAL LOGIC

1.1 Modal languages

A (unary, poly-)*modal similarity type* is a set τ of *modalities* $\alpha \in \tau$. Beside τ, we fix a (countable) set Φ of *propositional variables* or *atomic propositions.* With τ and Φ we associate the *modal language* $\mathrm{ML}(\tau, \Phi)$, in which every $\alpha \in \tau$ labels a modal *diamond operator* $\langle\alpha\rangle$. The formulae of $\mathrm{ML}(\tau, \Phi)$ are recursively defined as follows:

$$\varphi := \bot \mid p \mid (\varphi_1 \to \varphi_2) \mid \langle\alpha\rangle\varphi,$$

where $p \in \Phi$ and $\alpha \in \tau$, and unnecessary outer parentheses are dropped. The logical constant $\top$ and connectives $\neg, \wedge, \vee, \leftrightarrow$ may be introduced on an equal footing or are regarded as standard abbreviations. The operator $[\alpha]$, defined by $[\alpha]\,\varphi := \neg\langle\alpha\rangle\neg\varphi$, is the *box operator dual to* $\langle\alpha\rangle$. A formula not containing atomic propositions is called a *constant formula*.

To keep the notation simple, we regard the set Φ as fixed, and will usually not mention it explicitly. So we write $\mathrm{ML}(\tau)$, or also just ML when τ is clear from the context or irrelevant. We use the same notation for the set of all formulae of $\mathrm{ML}(\tau, \Phi)$, and in general identify notationally logical languages with their sets of formulae. In the *mono-modal* case of a modal similarity type consisting of a single unary modality, the only diamond and box are denoted by just $\Diamond$ and $\Box$, respectively.

DEFINITION 1. The *nesting depth* δ of a formula is defined recursively as follows:
$\delta(\bot) = \delta(p) = 0$;
$\delta(\varphi_1 \to \varphi_2) = \max(\delta(\varphi_1), \delta(\varphi_2))$;
$\delta(\langle\alpha\rangle\,\varphi) = \delta(\varphi) + 1$.
The fragment $\mathrm{ML}_n(\tau)$ comprises all formulae of $\mathrm{ML}(\tau)$ with nesting depth $\leq n$.

1.2 Kripke frames and structures

With the modal similarity type τ we associate a *relational similarity type* consisting of binary relations R_α for $\alpha \in \tau$. For simplicity we also denote this derived relational type by τ.

DEFINITION 2. A (*Kripke*) τ*-frame* is a relational τ-structure $\mathfrak{F} = \langle W, \{R_\alpha\}_{\alpha\in\tau}\rangle$ where $W \neq \emptyset$ and $R_\alpha \subseteq W \times W$ for each $\alpha \in \tau$. The domain W of $\mathfrak{F}$ is denoted by $\mathrm{dom}(\mathfrak{F})$. The relations $(R_\alpha)_{\alpha\in\tau}$ are the *accessibility* or *transition relations* in $\mathfrak{F}$. The elements of W, traditionally called *possible worlds*, will also be referred to, depending on the context, as *states, points, or nodes.* A *pointed* τ*-frame* is a pair $(\mathfrak{F}, w)$ where $w \in \mathrm{dom}(\mathfrak{F})$.

We also write $wR_\alpha u$ rather than $R_\alpha wu$ or $(w, u) \in R_\alpha$. Given a τ-frame $\mathfrak{F} = \langle W, \{R_\alpha\}_{\alpha\in\tau}\rangle$, every R_α defines two unary operators, $\langle R_\alpha\rangle$ and its dual $[R_\alpha]$, on $\mathcal{P}(W)$ as follows:

$$\langle R_\alpha\rangle\,(X) := \{w \in W \mid wR_\alpha u \text{ for some } u \in X\} \qquad \text{and} \qquad [R_\alpha](X) := \overline{\langle R_\alpha\rangle(\overline{X})}$$

where $\overline{X} := W \setminus X$ denotes the complement of X in W.

DEFINITION 3. A *Kripke structure* (*Kripke model*) over a τ-frame $\mathfrak{F} = \langle W, \{R_\alpha\}_{\alpha\in\tau}\rangle$ is a pair $\mathfrak{M} = \langle \mathfrak{F}, V\rangle$ where $V : \Phi \to \mathcal{P}(W)$ is a *valuation*, assigning to every atomic proposition p the set of states in W where p is declared true. The set W is the *domain of* $\mathfrak{M}$, denoted $\mathrm{dom}(\mathfrak{M})$. We often specify Kripke structures directly: $\mathfrak{M} = \langle W, \{R_\alpha\}_{\alpha\in\tau}, V\rangle$. A *pointed Kripke structure* is a pair $(\mathfrak{M}, w)$ where $w \in \mathrm{dom}(\mathfrak{M})$.

In any Kripke structure $\mathfrak{M} = \langle \mathfrak{F}, V\rangle$ the valuation V can be extended to a valuation of all formulae, which is again denoted by V. That extension is defined recursively as follows:[2]

$V(\bot) := \emptyset$;
$V(\varphi_1 \to \varphi_2) := \overline{V(\varphi_1)} \cup V(\varphi_2)$;
$V(\langle\alpha\rangle\,\varphi) := \langle R_\alpha\rangle\,(V(\varphi))$ (and $V([\alpha]\,\varphi) = [R_\alpha]\,(V(\varphi))$.

While first-order sentences express properties of a structure as a whole, modal formulae always make implicit reference to a distinguished (current) state in a Kripke structure. So the basic semantic notion in modal logic is *truth of a formula at a state of a Kripke structure*, with derived notions of validity also in Kripke structures and frames.

DEFINITION 4. A τ-formula φ is:

(i) *true at the state w of the τ-structure* $\mathfrak{M} = \langle \mathfrak{F}, V\rangle$, denoted $\mathfrak{M}, w \models \varphi$, if $w \in V(\varphi)$. This is the same as saying that φ is *true in the pointed structure* $(\mathfrak{M}, w)$. A formula that is true at a state of some τ-structure is *satisfiable.*

(ii) *valid in* $\mathfrak{M}$, denoted $\mathfrak{M} \models \varphi$, if $\mathfrak{M}, w \models \varphi$ for every $w \in \mathrm{dom}(\mathfrak{F})$, i.e., if $V(\varphi) = \mathrm{dom}(\mathfrak{F})$.

(iii) *(locally) valid at the state w of* $\mathfrak{F}$, denoted $\mathfrak{F}, w \models \varphi$, if $\mathfrak{M}, w \models \varphi$ for every τ-structure $\mathfrak{M}$ over $\mathfrak{F}$.
This is the same as saying that φ is *valid in the pointed frame* $(\mathfrak{F}, w)$.

(iv) *valid in* $\mathfrak{F}$, denoted $\mathfrak{F} \models \varphi$, if $\mathfrak{F}, w \models \varphi$ for every $w \in \mathrm{dom}(\mathfrak{F})$.
Equivalently: $\mathfrak{M} \models \varphi$ for every τ-structure $\mathfrak{M}$ over $\mathfrak{F}$.

(v) *valid*, denoted $\models \varphi$, if $\mathfrak{F} \models \varphi$ for every τ-frame $\mathfrak{F}$.

1.3 The standard translations into first- and second-order logic

With the modal language $\mathrm{ML}(\tau, \Phi)$, we associate the following purely relational vocabularies:

- the relational version of τ itself, consisting of R_α for $\alpha \in \tau$, and again denoted by just τ.
- the expansion τ_Φ of the relational vocabulary τ by unary predicates $\{P_0, P_1, \ldots\}$ associated with the atomic propositions $p_0, p_1, \ldots \in \Phi$.

Correspondingly, $\mathrm{FO}(\tau)$ and $\mathrm{FO}(\tau_\Phi)$ are the first-order languages with vocabularies τ and τ_Φ, respectively. We regard a τ-frame as a τ-structure in the usual sense, and a Kripke structure over a τ-frame as a τ_Φ-structure, with P_i interpreted as $V(p_i)$. We use the same notation for Kripke structures and for the associated first-order structures, as this causes no confusion. Wherever necessary, we will highlight the distinction by writing $\models_{\mathrm{FO}}$ to explicitly appeal to first-order semantics.

[2] In algebraic terms (see Chapter 6), the extended valuation is the unique homomorphism from the free τ-algebra of formulae to the modal algebra associated with the model $\mathfrak{M}$, extending V.

Truth and validity of a modal formula in a Kripke structure are *first-order notions* in the following sense. Let $\mathrm{VAR} = \{x_0, x_1, \ldots\}$ be the set of first-order variables of $\mathrm{FO}(\tau_\Phi)$. The formulae of $\mathrm{ML}(\tau)$ are translated into $\mathrm{FO}(\tau_\Phi)$ by means of the following *standard translation* [124, 127], parameterised with the variables from VAR:

- $\mathrm{ST}(p_i; x_j) := P_i x_j$ for every $p_i \in \Phi$;
- $\mathrm{ST}(\bot; x_j) := \bot$;
- $\mathrm{ST}(\varphi_1 \to \varphi_2; x_j) := \mathrm{ST}(\varphi_1; x_j) \to \mathrm{ST}(\varphi_2; x_j)$;
- $\mathrm{ST}(\langle\alpha\rangle\, \varphi; x_j) := \exists y(x_j R_\alpha y \wedge \mathrm{ST}(\varphi; y))$, where y is the first variable in $\mathrm{VAR} \setminus \{x_j\}$.

Note that only x_j is free in $\mathrm{ST}(\varphi; x_j)$. Furthermore, for the standard translation it suffices to use only the variables x_0 and x_1 (free or bound) in an alternating fashion. This yields a translation into the *two-variable fragment* FO^2 of first-order logic. Also, the standard translation of any modal formula falls into the *guarded fragment* of first-order logic. These observations are taken up in section 4.

The standard translation is semantically faithful in the following sense.

PROPOSITION 5. *For every pointed Kripke structure* $(\mathfrak{M}, w)$ *and* $\varphi \in \mathrm{ML}(\tau)$,

$$\mathfrak{M}, w \models \varphi \;\; \textit{iff} \;\; \mathfrak{M}, w \models_{\mathrm{FO}} \mathrm{ST}(\varphi; x_0).$$

While the semantics and validity for modal formulae over Kripke structures is thus essentially first-order, validity of a modal formula in a frame goes beyond first-order logic. Indeed, paraphrasing the definition in terms of the standard translation, a modal formula φ is valid in a frame iff its standard translation is true in that frame under *every* interpretation of the unary predicates occurring in it.

PROPOSITION 6. *For every pointed Kripke frame* $(\mathfrak{F}, w)$ *and* $\varphi \in \mathrm{ML}(\tau)$ *with atomic propositions among* $p_0, \ldots, p_n$*:*

$$\mathfrak{F}, w \models \varphi \;\; \textit{iff} \;\; \mathfrak{F}, w \models \forall P_0 \ldots \forall P_n \mathrm{ST}(\varphi; x_0).$$

Consequently, $\mathfrak{F} \models \varphi$ *iff* $\mathfrak{F} \models \forall P_0 \ldots \forall P_n \forall x_0 \mathrm{ST}(\varphi; x_0)$.

1.4 Theories, equivalence and definability

With every logic $\mathcal{L}$ comes an associated notion of logical equivalence between structures. Two structures of the appropriate type are equivalent with respect to $\mathcal{L}$ if no property expressible in $\mathcal{L}$ distinguishes between them, i.e., if their $\mathcal{L}$-theories are the same. In this sense, first-order logic gives rise to the notion of elementary equivalence. Correspondingly, *modal equivalence* is indistinguishability in modal logic. Each view of the semantics of modal logic – in terms of (pointed or plain) structures or frames – corresponds to a notion of modal theories and modal equivalence.

DEFINITION 7. The *modal theory* of a pointed Kripke τ-structure $(\mathfrak{M}, w)$ is the set of all formulae of $\mathrm{ML}(\tau)$ satisfied in $(\mathfrak{M}, w)$: $\mathrm{Th}_{\mathrm{ML}}(\mathfrak{M}, w) := \{\varphi \in \mathrm{ML}(\tau) \mid \mathfrak{M}, w \models \varphi\}$.

Correspondingly, the modal theory of $\mathfrak{M}$ is $\mathrm{Th}_{\mathrm{ML}}(\mathfrak{M}) := \{\varphi \in \mathrm{ML}(\tau) \mid \mathfrak{M} \models \varphi\}$. The modal theories of a frame and pointed frame, as well as of classes of (pointed) Kripke structures or frames, are defined likewise.

The basic notion of modal equivalence, corresponding to the notion of truth at a state of a Kripke structure, is an equivalence relation on the class of *pointed Kripke structures* $(\mathfrak{M}, w)$. Natural variants cover the derived notions for plain Kripke structures, and for pointed or plain frames.

DEFINITION 8. For two pointed Kripke τ-structures $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$: $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ are ML-*equivalent*, denoted $(\mathfrak{M}, w) \equiv_{\mathrm{ML}} (\mathfrak{M}', w')$, iff they satisfy exactly the same formulae of ML, i.e., iff $\mathrm{Th}_{\mathrm{ML}}(\mathfrak{M}, w) = \mathrm{Th}_{\mathrm{ML}}(\mathfrak{M}', w')$. Modal equivalence between Kripke structures, frames, and pointed frames are defined likewise.

Definability in modal logic means different things corresponding to the different levels of the semantics. We distinguish local versus global definability (truth at a state versus validity throughout a frame/structure), and definability at the level of structures versus frames (truth/validity for a given valuation versus for all valuations).

Given a formula $\varphi \in \mathrm{ML}(\tau)$, the classes of pointed Kripke structures, Kripke structures, pointed frames and frames defined by φ are denoted as $\mathrm{KS}(\varphi)$, $\mathrm{PKS}(\varphi)$, $\mathrm{FR}(\varphi)$, and $\mathrm{PFR}(\varphi)$, respectively:

$$\mathrm{PKS}(\varphi)=\{(\mathfrak{M}, w) \mid \mathfrak{M}, w \models \varphi\} \qquad \mathrm{PFR}(\varphi)=\left\{(\mathfrak{F}, w) \;\middle|\; \begin{array}{l}(\mathfrak{F}, V), w \models \varphi \\ \text{for all valuations } V\end{array}\right\}$$

$$\mathrm{KS}(\varphi)=\{\mathfrak{M} \mid \mathfrak{M} \models \varphi \text{ for all } w \in \mathrm{dom}(\mathfrak{M})\} \qquad \mathrm{FR}(\varphi)=\left\{\mathfrak{F} \;\middle|\; \begin{array}{l}(\mathfrak{F}, V), w \models \varphi \text{ for all } w \\ \text{and for all valuations } V\end{array}\right\}$$

DEFINITION 9. A class $\mathcal{P}$ of pointed Kripke τ-structures is *(modally) definable* in the language $\mathrm{ML}(\tau)$ if $\mathcal{P} = \mathrm{PKS}(\varphi)$ for some formula $\varphi \in \mathrm{ML}(\tau)$. Definable classes of Kripke structures, frames, and pointed frames are defined likewise.

EXAMPLE 10. Here are some examples of modally definable classes of Kripke frames and structures.

The class of pointed Kripke structures $(\mathfrak{M}, w)$, where $\mathfrak{M} = \langle W, R, V\rangle$, such that w has at least one successor not satisfying p for which every successor satisfies q, is defined by the formula $\Diamond(\neg p \wedge \Box q)$.

The formula $p \rightarrow \Box p$ defines the class of Kripke structures in which the valuation of p is closed under the accessibility relation.

The class of frames in which every state has a successor is defined by the formula $\Diamond\top$; the same formula defines the class of pointed frames $(\mathfrak{F}, w)$ in which w has a successor.

The formula $\Diamond p \rightarrow \Box p$ defines the class of frames $\mathcal{K}$ in which every state has at most one successor. It is straightforward to show that the formula is valid in every such frame. For the converse: if the formula fails at some state w of a Kripke structure over a frame $\mathfrak{F}$, then p is true at some successor of w. But since $\Box p$ is false at w, there must be another successor of w where p fails. Hence $\mathfrak{F}$ does not satisfy the defining property of $\mathcal{K}$.

Other standard examples of modally definable classes of frames include the classes of: reflexive frames, defined by $\Box p \rightarrow p$; transitive frames, defined by $\Box p \rightarrow \Box\Box p$; symmetric frames, defined by $\Diamond\Box p \rightarrow p$, etc. For more examples see [117, 75, 127, 128].

Proposition 5 implies that the definability of classes of (pointed) Kripke structures by modal formulae is a special case of first-order definability. Consequently, modal logic shares many basic model-theoretic results with first-order logic, such as compactness and Löwenheim–Skolem theorems (see [12, 68]). We will discuss the model theoretic aspects of modal logic as a fragment of first-order logic on Kripke structures in section 4.

On the other hand, Proposition 6 indicates that modal definability of (pointed) frames is a form of Π^1_1-definability, and the model-theoretic consequences of that fact will be discussed in section 8. In particular, we will see that it is indeed essentially second-order.

1.5 Polyadic modalities

In polyadic modal logics one considers modalities α of arbitrary arities $r(\alpha) \in \mathbb{N}$, which give rise to formulae $\langle\alpha\rangle(\varphi_1, \ldots, \varphi_n)$ if $n = r(\alpha)$. The interpretation of an n-ary modal operator α is given in terms of $(n+1)$-ary relations R_α in corresponding frames, and an n-ary operator on subsets of these frames, in such a way that the semantics is faithfully captured in the standard translation $\mathrm{ST}(\langle\alpha\rangle(\varphi_1, \ldots, \varphi_n; x_j)) := \exists y_1 \ldots \exists y_n (x_j R_\alpha y_1 \ldots y_n \wedge \bigwedge_{i=1}^n \mathrm{ST}(\varphi_i; y_i))$, where $y_1 \ldots y_n$ are the first n variables in $\mathrm{VAR} \setminus \{x_j\}$ (and $x_j R_\alpha y_1 \ldots y_n$ is just a notational variant for $R_\alpha x_j y_1 \ldots y_n$).

Polyadic modalities were first studied from an algebraic perspective, as normal and additive operators in Boolean algebras, by Jónsson and Tarski [78, 79]. All the essential model theoretic features of modal logic can be generalised to this more liberal setting, albeit with some care and sometimes unavoidable notational complications. In [41] Goguadze et al define and develop systematically an interpretation of polyadic languages into monadic ones, and simulations of polyadic by monadic logics, which transfer a number of important properties, such as frame completeness, finite model property, canonicity and first-order definability. On the other hand, so called *purely modal polyadic languages* are defined in [55], where all logical connectives except negation are treated as binary modalities, and modalities can be composed. Thus, all polyadic modal formulae are built from (composite) boxes and diamonds applied to literals, making their syntactic structure much simpler.

Throughout this chapter we will only treat monadic modalities explicitly.

2 BISIMULATION AND BASIC MODEL CONSTRUCTIONS

A major concern in model theory is the analysis of logical equivalence of structures in comparison with other natural notions of structural equivalence, in particular equivalences of a more combinatorial or algebraic nature. Bisimulation equivalences as studied below prove to be the algebraic/combinatorial counterparts to modal equivalence.

For first-order logic this combinatorial approach leads to the well-known characterisation of elementary equivalence via *Ehrenfeucht–Fraïssé games* (see [68, 108, 26, 25]). Variations of the basic Ehrenfeucht–Fraïssé idea apply to many other logics including modal logic. Modal equivalence can thus be put into the general Ehrenfeucht–Fraïssé framework. We shall sketch this connection in section 4. The very natural game associated with modal equivalence has, however, also been invented and studied independently and in its own right, with the notions of *zig-zag relation* (van Benthem) and *bisimulation equivalence* (Hennessy, Milner, Park). We therefore put an autonomous, modal treatment before the discussion of relationships with the general framework of Ehrenfeucht–Fraïssé and pebble games.

2.1 Bisimulation and invariance

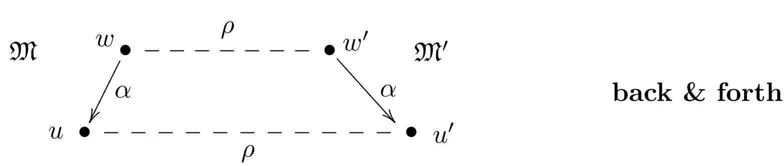
While the notion of logical equivalence is static, it can often be characterised in more dynamic, game-theoretic terms. The concept of bisimulation equivalence, which is closely related to corresponding games, is one of the most productive ideas in the model theory of modal logics, temporal logics, logics for concurrency, etc. Just as it has multiple roots in these various branches of logic, many variants have been employed to capture specific notions of "behavioural equivalence" between all kinds of transition systems that are interesting in their own right for various application areas – and not necessarily with any 'logic' in mind.

DEFINITION 11. Let $\mathfrak{M} = \langle W, \{R_\alpha\}_{\alpha\in\tau}, V\rangle$ and $\mathfrak{M}' = \langle W', \{R'_\alpha\}_{\alpha\in\tau}, V'\rangle$ be two Kripke τ-structures. A *bisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$ is a non-empty relation $\rho \subseteq W \times W'$ satisfying the following conditions for any $w\rho w'$:

Atom equivalence: w and w' satisfy the same atomic propositions, hereafter denoted by $w \simeq w'$.

Forth: For any $\alpha \in \tau$, if $wR_\alpha u$ for some $u \in W$, then there is some $u' \in W'$ such that $w'R'_\alpha u'$ and $u\rho u'$. (Any α-transition at w in $\mathfrak{M}$ can be matched at w' in $\mathfrak{M}'$.)

Back: Similarly, in the opposite direction: for any $\alpha \in \tau$, and $w'R'_\alpha u'$ there is some $u \in W$ such that $wR_\alpha u$ and $u\rho u'$. (Any α-transition at w' in $\mathfrak{M}'$ can be matched at w in $\mathfrak{M}$.)



That ρ is a bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$ is denoted as $\rho\colon \mathfrak{M} \rightleftarrows \mathfrak{M}'$. If, moreover, ρ is such that every element in $\mathfrak{M}$ is linked to some element of $\mathfrak{M}'$ and vice versa, we say that ρ is a *global bisimulation* and that $\mathfrak{M}$ and $\mathfrak{M}'$ are *globally bisimilar*.

DEFINITION 12. Two pointed Kripke structures $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ are *bisimilar* or *bisimulation equivalent*, denoted $(\mathfrak{M}, w) \rightleftarrows (\mathfrak{M}', w')$, if there is a bisimulation ρ between $\mathfrak{M}$ and $\mathfrak{M}'$ such that $w\rho w'$.

Bisimulations between (pointed) frames can be defined likewise, by omitting *atom equivalence*. Thus, a relation ρ is a bisimulation between two frames $\mathfrak{F}$ and $\mathfrak{F}'$, iff it is a bisimulation between the respective Kripke structures $\langle\mathfrak{F}, V_\perp\rangle$ and $\langle\mathfrak{F}', V'_\perp\rangle$ where the valuations $V_\perp$ and $V'_\perp$ render every atomic proposition false at every state of the respective frame.

DEFINITION 13. Let $\mathcal{C}$ be a class of structures appropriate for the logical language $\mathcal{L}$ (e.g., pointed Kripke structures for ML). Let $\approx$ be an equivalence relation on $\mathcal{C}$. Then $\mathcal{L}$ is *preserved under* $\approx$ over $\mathcal{C}$, or $\mathcal{L}$ is $\approx$*-invariant* over $\mathcal{C}$, iff for any $\mathfrak{A} \approx \mathfrak{A}'$ in $\mathcal{C}$ and any $\varphi \in \mathcal{L}$: $\mathfrak{A} \models \varphi \;\Leftrightarrow\; \mathfrak{A}' \models \varphi$, i.e., $\mathfrak{A}$ and $\mathfrak{A}'$ are $\mathcal{L}$-equivalent. In other words: $\approx$ is a refinement of $\equiv_\mathcal{L}$, or $\approx\, \subseteq\, \equiv_\mathcal{L}$.

Invariance phenomena give insights into the semantics of the logic involved, and also often provide key tools for the model theoretic study of the logic (e.g., model constructions guided by $\approx$ equivalence). The relationship between modal logics and bisimulation equivalences provides an excellent example of such a fruitful companionship.

It would be straightforward to prove the following by induction on the structure of modal formulae, straight from the definition of bisimulations. However, this will also fall out as a corollary of the more instructive analysis of the associated bisimulation games. We therefore meanwhile only state the fact.

THEOREM 14 (bisimulation invariance).
$\mathrm{ML}(\tau)$ *is bisimulation invariant: if* $(\mathfrak{M}, w) \rightleftarrows (\mathfrak{M}', w')$, *then* $(\mathfrak{M}, w) \equiv_{\mathrm{ML}} (\mathfrak{M}', w')$.

Consequently, for every constant formula $\theta \in \mathrm{ML}(\tau)$ and pointed τ-frames $(\mathfrak{F}, w)$ and $(\mathfrak{F}', w')$: if $(\mathfrak{F}, w) \rightleftarrows (\mathfrak{F}', w')$, then $(\mathfrak{F}, w) \models \theta$ iff $(\mathfrak{F}', w') \models \theta$.

2.2 Classical truth-preserving constructions

Bisimulations induced by maps from one frame to another have classically been studied as *bounded morphisms* or *p-morphisms.* We state the corresponding back-and-forth conditions, which are slightly simpler in the case of such a functional relationship, and treat some particularly important special cases. The use of *generated and rooted substructures*, *bounded morphic images*, *tree unfoldings* and *disjoint unions* in connection with classical model constructions for modal logic is based on truth preservation for modal formulae. These constructions were introduced for basic modal logic [117, 7] before the notion of bisimulation was developed and its importance for modal logic realised. Via *duality theory*, which connects the relational semantics for modal logic with an algebraic semantics, bounded morphisms, generated subframes and disjoint unions correspond respectively to the fundamental universal algebraic notions of subalgebras, homomorphic images, and direct products. For details see Chapter 6 of this handbook, as well as [78, 43, 44, 114] and [5, Ch. 5].

The preservation results encountered in these special cases of a passage to bisimilar structures highlight to various degrees one of the key characteristic features of the semantics of modal logic: its *explicit locality* and *restricted nature of quantification.* Unlike first-order logic, whose global quantification over the entire universe makes truth generally dependent on the entire structure, the truth of a modal formula in a Kripke structure is evaluated relative to a 'current' state and admits access to the rest of the structure only along the edges of the accessibility relations.

Passage from a given structures to a bisimilar tree structure, obtained via a simple bounded morphism, shows for instance that any satisfiable formula of basic modal logic is satisfied at the root of a tree structure (tree model property, see Corollary 24; this can be further strengthened to a finite tree model property, see Lemma 35). Conversely, preservation results can be used to show that certain properties are not modally definable. We shall see some classical examples of this in section 2.3.

Bounded morphisms

DEFINITION 15. Let $\mathfrak{M} = \langle W, \{R_\alpha\}_{\alpha\in\tau}, V\rangle$ and $\mathfrak{M}' = \langle W', \{R'_\alpha\}_{\alpha\in\tau}, V'\rangle$ be Kripke structures. A function $\rho\colon W \to W'$ is a *bounded morphism* from $\mathfrak{M}$ to $\mathfrak{M}'$ if its graph is a bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$. We denote a bounded morphism as in $\rho\colon \mathfrak{M} \xrightarrow{\rightleftarrows} \mathfrak{M}'$.

Bounded morphisms between frames are similarly defined.

If ρ is onto, then $\mathfrak{M}'$ is a *bounded morphic image of* $\mathfrak{M}$ (and similarly for frames).

Thus, for each $u \in W$, a bounded morphism ρ uniquely singles out a bisimilar state $\rho(w)$ in W'. The bisimulation conditions for a bounded morphism between two Kripke structures correspondingly become:

Atom equivalence: $w \simeq \rho(w)$ for every $w \in W$.

Forth: For any $w \in W$ and $\alpha \in \tau$, if $wR_\alpha u$ for some $u \in W$, then $\rho(w)R'_\alpha\rho(u)$.

Back: For any $w \in W$ and $\alpha \in \tau$, if $\rho(w)R'_\alpha u'$ for some $u' \in W'$, then $u' = \rho(u)$ for some $u \in W$ such that $wR_\alpha u$.

Bisimulation invariance yields the following preservation results.

COROLLARY 16. *Bounded morphisms preserve truth and validity of modal formulae. More specifically, if* $\rho\colon \mathfrak{M} \overset{\rightleftarrows}{\longrightarrow} \mathfrak{M}'$ *is a bounded morphism and* $\varphi \in \mathrm{ML}(\tau)$*, then:*

(i) *for all* $u \in \mathrm{dom}(\mathfrak{F})$*:* $\mathfrak{M}, u \models \varphi$ *iff* $\mathfrak{M}', \rho(u) \models \varphi$.

(ii) *If* ρ *is onto, then* $\mathfrak{M} \models \varphi$ *iff* $\mathfrak{M}' \models \varphi$*, i.e.,* $\mathrm{Th}_{\mathrm{ML}}(\mathfrak{M}) = \mathrm{Th}_{\mathrm{ML}}(\mathfrak{M}')$.

(iii) *If* $\mathfrak{F}, u \models \varphi$*, then* $\mathfrak{F}', \rho(u) \models \varphi$.

(iv) *If* ρ *is onto, then* $\mathfrak{F} \models \varphi$ *implies* $\mathfrak{F}' \models \varphi$.

For the latter two claims one just has to note that each model $\mathfrak{M}' = \langle \mathfrak{F}', V' \rangle$ over the frame $\mathfrak{F}'$ can be pulled back to give a model $\mathfrak{M} = \langle \mathfrak{F}, V \rangle$ over the frame $\mathfrak{F}$ via $V(p) := \rho^{-1}[V'(p)] = \{w \in \mathrm{dom}(\mathfrak{F}) \mid \rho(w) \in V'(p)\}$. This turns ρ into a bounded morphism from $\mathfrak{M}$ to $\mathfrak{M}'$. Note, however, that not every model over $\mathfrak{F}$ is obtained in this manner.

We turn to several basic model constructions involving bounded morphisms: generated substructures, rooted substructures, tree unfoldings and disjoint unions.

Generated and rooted substructures

If $R \subseteq W^2$ is any binary relation over W, and $W' \subseteq W$, we write $R{\restriction}W'$ for the restriction of R to W', $R{\restriction}W' = R \cap (W' \times W')$. Similarly for a valuation V on W, $V{\restriction}W'$ stands for its restriction to W'.

DEFINITION 17. Let $\mathfrak{F} = \langle W, \{R_\alpha\}_{\alpha\in\tau}\rangle$ be a frame, or $\mathfrak{M} = \langle \mathfrak{F}, V \rangle$ a Kripke structure over $\mathfrak{F}$, respectively, and $W' \subseteq W$.

(i) The induced *subframe of* $\mathfrak{F}$ *over* W' is the frame $\mathfrak{F}' := \mathfrak{F}{\restriction}W' = \langle W', \{R_\alpha{\restriction}W'\}_{\alpha\in\tau}\rangle$. The subframe relationship is denoted $\mathfrak{F}' \leq \mathfrak{F}$.

(ii) $\mathfrak{F}' = \mathfrak{F}{\restriction}W'$ is a *generated subframe of* $\mathfrak{F}$, denoted $\mathfrak{F}' \trianglelefteq \mathfrak{F}$, if W' is closed under all accessibility relations in the sense that $wR_\alpha u$ for $w \in W'$ implies $u \in W'$.

(iii) The induced *substructure of* $\mathfrak{M}$ *over* W' is the Kripke structure $\mathfrak{M}' = \mathfrak{M}{\restriction}W' = \langle \mathfrak{F}{\restriction}W', V{\restriction}W'\rangle$, denoted $\mathfrak{M}' \leq \mathfrak{M}$. If $\mathfrak{F}{\restriction}W' \trianglelefteq \mathfrak{F}$, then $\mathfrak{M}'$ is a *generated substructure of* $\mathfrak{M}$, denoted $\mathfrak{M}' \trianglelefteq \mathfrak{M}$.

Obviously, for $\mathfrak{M}' \trianglelefteq \mathfrak{M}$ the inclusion map $\rho\colon W' \to W$ is a bounded morphism. By bisimulation invariance, we therefore have the following.

PROPOSITION 18. *For all Kripke structures* $\mathfrak{M}' \trianglelefteq \mathfrak{M}$ *and for every formula* φ *of* $ML(\tau)$*:*

(i) *for every* $u \in \mathrm{dom}(\mathfrak{M}') : \mathfrak{M}, u \models \varphi$ *iff* $\mathfrak{M}', u \models \varphi$.

(ii) $\mathfrak{M} \models \varphi$ *implies* $\mathfrak{M}' \models \varphi$.

Likewise, for frames $\mathfrak{F}' \trianglelefteq \mathfrak{F}$ *and* $u \in \mathrm{dom}(\mathfrak{F}')$*:* $\mathfrak{F}, u \models \varphi$ *iff* $\mathfrak{F}', u \models \varphi$*, and* $\mathfrak{F} \models \varphi$ *implies* $\mathfrak{F}' \models \varphi$.

The latter claim holds since every Kripke structure over $\mathfrak{F}'$ is induced by a Kripke structure on $\mathfrak{F}$.

A particularly important case of generated subframes deals with the set of all states reachable from a fixed state. A *path* in a frame $\mathfrak{F} = \langle W, \{R_\alpha\}_{\alpha\in\tau}\rangle$ is a sequence $\vec{w} = (w_0, \alpha_1, w_1, \ldots, \alpha_k, w_k)$, where $w_{i-1} R_{\alpha_i} w_i$ for $i = 1, \ldots, k$ (this path is *rooted* at w_0 and has *length* k). A path of length $k = 0$, $\vec{w} = (w_0)$, is identified with its root w_0. For $u \in W$, we denote the set of all paths rooted at u by $\vec{W}[u]$. For every path $\vec{w}$ as above we define the 'terminal state' function $f(\vec{w}) = w_k$ where k is the length of $\vec{w}$. Then

$$W[u] := \{f(\vec{w}) \mid \vec{w} \in \vec{W}[u]\}$$

is the set of all states in $\mathfrak{F}$ reachable from u (including u itself).

DEFINITION 19. Let $\mathfrak{F} = \langle W, \{R_\alpha\}_{\alpha\in\tau}\rangle$ be a frame, $\mathfrak{M} = \langle \mathfrak{F}, V\rangle$ a Kripke structure over $\mathfrak{F}$, and $u \in W$.

(i) The *subframe of* $\mathfrak{F}$ *rooted at* u is the frame $\mathfrak{F}[u] = \mathfrak{F}\restriction W[u]$.

(ii) The *substructure of* $\mathfrak{M}$ *rooted at* u is the Kripke structure $\mathfrak{M}[u] = \mathfrak{M}\restriction W[u]$.

(iii) $\mathfrak{F}$ (respectively $\mathfrak{M}$) is *rooted at* u if $W[u] = W$.

Clearly, for any $u \in W$: $\mathfrak{F}[u] \trianglelefteq \mathfrak{F}$ and $\mathfrak{M}[u] \trianglelefteq \mathfrak{M}$, respectively. Therefore, we obtain the following.

COROLLARY 20. *For every Kripke structure* $\mathfrak{M}' \trianglelefteq \mathfrak{M}$ *and formula* φ *of* $ML(\tau)$*:*

(i) *for all* $u \in W$*:* $\mathfrak{M}, u \models \varphi$ *iff* $\mathfrak{M}[u], u \models \varphi$.

(ii) $\mathfrak{M} \models \varphi$ *implies* $\mathfrak{M}[u] \models \varphi$.

(iii) *Likewise for (pointed) frames.*

Thus, any satisfiable formula is satisfiable at the root of a rooted Kripke structure.

Tree unfoldings

An important model construction based on a canonical bounded morphism is the *unfolding* or *tree unravelling* of a Kripke structure $\mathfrak{M} = \langle W, \{R_\alpha\}_{\alpha\in\tau}, V\rangle$ from some $u \in W$. This construction was introduced in [113], where the tree model property (cf. Corollary 24 below) was proved, too.

Recall the map $f\colon \vec{W}[u] \to W$, which maps the path $\vec{w} = (u = w_0, \alpha_1, w_1, \ldots, \alpha_k, w_k)$ to its terminal state $f(\vec{w}) = w_k$. The unfolding of $\vec{\mathfrak{M}}[u]$ of $\mathfrak{M}$ at u is based on the set $\vec{W}[u]$ of all paths rooted at u, with the natural definition of accessibility relations and a valuation that turns f into a bounded morphism.

DEFINITION 21. The *unfolding* (or, *unravelling*) of $\mathfrak{M} = \langle W, \{R_\alpha\}_{\alpha\in\tau}, V\rangle$ from some $u \in W$ is the rooted Kripke structure $\vec{\mathfrak{M}}[u] := \langle \vec{W}[u], \{\vec{R}_\alpha\}_{\alpha\in\tau}, \vec{V}\rangle$ with root $u = (u)$, where

$$\begin{array}{rcl} \vec{R}_\alpha & := & \{(\vec{w}, (\vec{w}, \alpha, w')) \mid \vec{w} \in \vec{W}[u], f(\vec{w}) R_\alpha w'\}, \\ \vec{V}(p) & := & f^{-1}[V(p)]. \end{array}$$

Indeed, $\vec{\mathfrak{M}}[u]$ is a tree structure with root $u = (u)$ in the sense of the following definition.

DEFINITION 22. A pointed frame $\mathfrak{F} = \langle W, \{R_\alpha\}_{\alpha\in\tau}\rangle$ with distinguished state $u \in W$ is a *tree* with root u if $\mathfrak{F}$ is rooted at u and every state $w \in W$ is reachable from u by a unique path. Accordingly, every Kripke structure over $(\mathfrak{F}, u)$ is a tree structure.

OBSERVATION 23. For every pointed Kripke structure $(\mathfrak{M}, u)$ the terminal state map $f\colon \vec{W}[u] \to W[u]$ is a bounded morphism of the unfolding $\vec{\mathfrak{M}}[u]$ onto $\mathfrak{M}[u]$.

As $\vec{\mathfrak{M}}[u]$ is a tree structure with root $u = (u)$, we obtain the following. Also compare Lemma 35 below.

COROLLARY 24 (tree-model property). *Every satisfiable modal formula is satisfiable at the root of a tree.*

Disjoint unions

Disjoint unions are well known for relational structures: the component structures are put side by side without any relational links between the components. Assuming that the given family of Kripke structures or frames is based on universes that are pairwise disjoint, we may just take the set-theoretic union of the universes, accessibility relations, and valuations, respectively. If the given frames are not disjoint, they first need to be replaced by isomorphic copies over universes that are pairwise disjoint.

To be specific, define the *disjoint union* of an arbitrary family $\{W^i\}_{i\in I}$ of (not necessarily disjoint) sets as $\biguplus_{i\in I} W^i := \bigcup_{i\in I}(W^i \times \{i\})$. With this formalisation, we have the natural injection or embedding $\varepsilon_j\colon W^j \to \bigcup_{i\in I}(W^i \times \{i\})$ of each component set into the disjoint union, which maps $w \in W^j$ to $(w, j) \in \bigcup_{i\in I}(W^i \times \{i\})$.

DEFINITION 25. Consider a family of τ-frames $\{\mathfrak{F}^i = \langle W^i, \{R^i_\alpha\}_{\alpha\in\tau}\rangle\}_{i\in I}$ and a family of Kripke structures $\{\mathfrak{M}^i = \langle\mathfrak{F}^i, V^i\rangle\}_{i\in I}$ over these.

(i) The *disjoint union of* $\{\mathfrak{F}^i\}_{i\in I}$ is the frame $\biguplus_{i\in I}\mathfrak{F}^i = \langle\biguplus_{i\in I} W^i, \{R_\alpha\}_{\alpha\in\tau}\rangle$, where $(w_0, i_0)R_\alpha(w_1, i_1)$ iff $i_0 = i_1 = i$ and $w_0 R^i_\alpha w_1$.

(ii) The *disjoint union of* $\{\mathfrak{M}^i\}_{i\in I}$ is the Kripke τ-structure $\biguplus_{i\in I}\mathfrak{M}^i = \langle\biguplus_{i\in I}\mathfrak{F}^i, V\rangle$ where $V(p) = \biguplus_{i\in I} V^i(p)$.

It is immediate that the natural injection $\varepsilon_j\colon W^j \to \biguplus_{i\in I} W^i$ isomorphically embeds $\mathfrak{M}^j$ into $\biguplus_{i\in I}\mathfrak{M}^i$ and is indeed a bounded morphism with image $\epsilon_j[\mathfrak{M}^j] \simeq \mathfrak{M}^j$ and $\epsilon_j[\mathfrak{M}^j] \trianglelefteq \biguplus_{i\in I}\mathfrak{M}^i$. We therefore obtain the following, by bisimulation invariance and based on previous observations.

PROPOSITION 26. *Given a family of τ-frames $\{\mathfrak{F}^i = \langle W^i, \{R^i_\alpha\}_{\alpha\in\tau}\rangle\}_{i\in I}$, a family of Kripke structures $\{\mathfrak{M}^i = \langle\mathfrak{F}^i, V^i\rangle\}_{i\in I}$ over these frames, and $\varphi \in \mathrm{ML}(\tau)$:*

(i) *For every $j \in I$ and $w \in \mathrm{dom}(\mathfrak{M}^j)$: $\mathfrak{M}^j, w \models \varphi$ iff $\biguplus_{i\in I}\mathfrak{M}^i, (w, j) \models \varphi$.*

(ii) *For every $j \in I$ and $w \in \mathrm{dom}(\mathfrak{F}^j)$: $\mathfrak{F}^j, w \models \varphi$ iff $\biguplus_{i\in I}\mathfrak{F}^i, (w, j) \models \varphi$.*

(iii) *$\biguplus_{i\in I}\mathfrak{M}^i \models \varphi$ iff $\mathfrak{M}^i \models \varphi$ for every $i \in I$.*

(iv) *$\biguplus_{i\in I}\mathfrak{F}^i \models \varphi$ iff $\mathfrak{F}^i \models \varphi$ for every $i \in I$.*

The following structural observation [127] links some of the ideas explored in this section.

PROPOSITION 27. *Any Kripke structure is the bounded morphic image of a disjoint union of rooted Kripke structures, and indeed of tree structures.*

For a proof of the proposition, consider the families of the $\{\mathfrak{M}[u]\}_{u \in W}$ or $\{\vec{\mathfrak{M}}[u]\}_{u \in W}$. The desired bounded morphisms (from the disjoint unions of these families back onto $\mathfrak{M}$) are the unions of the projection and inclusion or terminal state maps defined on the components of these disjoint unions.

2.3 Proving non-definability

To show that a given property of structures is definable in a given logic, it suffices simply to find a defining formula. Showing that a property is *not* definable, however, is not so straightforward, and often requires elaborate arguments. A standard method for establishing non-definability of a property $\mathcal{P}$ (i.e., of the class of structures satisfying that property) in a logic $\mathcal{L}$ is to show that $\mathcal{P}$ is not closed under some construction preserving truth (validity) of all formulae of $\mathcal{L}$. Now that we have at hand constructions that preserve truth and validity of modal formulae, we can use them to show that various properties of frames and structures are not modally definable. Compare Definition 9 for the relevant notions of definability.

At the level of pointed Kripke structures, modal formulae capture only properties that are local in the sense that whether or not $\mathfrak{M}, w \models \varphi$ only depends on $(\mathfrak{M}[w], w)$. In other words, modal formulae are incapable of expressing any property of $(\mathfrak{M}, w)$ that involves points beyond $\mathfrak{M}[w]$. For instance, there is no $\varphi \in$ ML such that $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M} \models p$. Indeed, one can always add to $\mathfrak{M}$ an extra point (as a disjoint union), not reachable from w, where p is false. The resulting pointed structure $(\mathfrak{M}', w)$ is bisimilar to $(\mathfrak{M}, w)$, whence φ would have to be equally true or false at w in both.

Likewise, at the level of Kripke structures, there is no $\varphi \in$ ML such that $\mathfrak{M} \models \varphi$ iff the accessibility relation of the underlying frame $\mathfrak{F}$ is reflexive. This follows for instance from the fact that the unfolding of any frame is irreflexive. If $\mathfrak{M}$ is reflexive, then so is the generated substructure $\mathfrak{M}[u]$, which however is also a bounded morphic image of the irreflexive $\vec{\mathfrak{M}}[u]$. Reflexivity, however, is well-known to be definable in terms of frame validity by the formula $\Box p \to p$. In other words, the class of reflexive frames is definable by the second-order sentence $\forall P \forall x(\forall y(Rxy \to Py) \to Px)$. Intuitively, in terms of truth in Kripke structures, modal formulae can make very little reference to the underlying frame.

We turn to properties of frames and non-definability in terms of frame validity, which is maybe the most interesting facet of modal expressiveness. One can show, for instance, that (unlike reflexivity) *irreflexivity* is not a modally definable frame property. This property is not preserved under surjective bounded morphisms, while surjective bounded morphisms preserve frame validity. One may consider unfoldings as above, or also the (irreflexive) frame $\mathfrak{F} = \langle \{w_1, w_2\}, \{(w_1, w_2), (w_2, w_1)\} \rangle$ and its bounded morphic image $\mathfrak{F}' = \langle \{w\}, \{(w, w)\} \rangle$, which is reflexive. Hence any φ valid in the former would also be valid in the latter.

Similarly, the class of non-reflexive frames (i.e., ones having at least one irreflexive point) is not definable in terms of validity of modal formulae, because it is not closed under passage to generated subframes. Likewise, the classes of finite frames, connected frames, or of frames with a universal accessibility relation ($R = W^2$), are not definable in terms of frame validity of modal formulae, as they are not closed under disjoint unions.

For another interesting example, consider the property of a frame to be a reflexive partial ordering. It is not modally definable, because anti-symmetry is not preserved under surjective bounded morphisms. Indeed, $\langle \mathbb{Z}, \leq \rangle$ is antisymmetric, but the mapping of it onto the symmetric frame $\mathfrak{F}$ above, sending all odd numbers to w_1 and all even ones to w_2, is a surjective bounded morphism (and remains so, even when we add an inverse or past modality, as in basic temporal logic).

However, the preservation results we have discussed so far are insufficient to capture frame non-definability. A witness is the following more subtle example: the property of *continuity,* or *Dedekind completeness* is not modally definable in modal logic, but to see that using a non-preservation argument is not easy. Ultimately, this follows from the fact that $\langle \mathbb{R}, \leq \rangle$ (which is continuous) and $\langle \mathbb{Q}, \leq \rangle$ (which is not) have the same *modal theory* (i.e., the same formulae of basic modal logic are valid in these frames); see [46].[3]

Finally, note that preservation under generated subframes, surjective bounded morphisms and disjoint unions is *not* sufficient to guarantee modal definability in terms of frame validity, even for first-order definable properties. For instance, the class of frames defined by the first-order sentence $\forall x \exists y (xRy \wedge yRy)$ (see [51, 128, 74]) is not modally definable, despite being closed under these three constructions. We will come back to this example in section 6.1.

For more examples of modal non-definability see [5, Section 3.3] and [128] where syntactic characterisations of the first-order properties preserved by each one of the three constructions mentioned above have been obtained.

3 BISIMULATION: A CLOSER LOOK

3.1 Bisimulation games

Bisimulation relations may be understood as descriptions of (non-deterministic) winning strategies for one player in corresponding model comparison games. We illustrate the concept in the case of bisimulations for basic modal logic – or Kripke structures with a single binary transition relation R – writing $\Diamond$ and $\Box$ for the associated modalities. All considerations admit canonical ramifications to the more general poly-modal setting (as well as to polyadic modalities).

Let $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$ be Kripke structures of this basic type. The *bisimulation game* over $\mathfrak{M}$ and $\mathfrak{M}$ is played by two players **I** and **II** with one pebble in $\mathfrak{M}$ and one in $\mathfrak{M}'$ to mark a single 'current' state in each structure. A configuration in the game consists of a current placement of the two pebbles and is described by the pair of pointed Kripke structures $(\mathfrak{M}, w; \mathfrak{M}', w')$, with distinguished w and w' for the current states (pebble positions).

A single *round* in the game is played as follows. The first player, **I**, or *challenger*, selects one of the two pebbles and moves it forward along an edge in the respective structure to a successor state. The second player, **II**, or *defender*, has to respond by similarly moving forward the pebble in the opposite structure.

[3] On the other hand, continuity is definable in temporal logic by the formula $\Box([P]p \to \langle F \rangle [P]p) \to ([P]p \to [F]p)$, where F and P are respectively the *future* and *past* modality, and $\Box\varphi = [P]\varphi \wedge \varphi \wedge [F]\varphi$ is the *always* modality. See [46].

During the game, **II** loses when no such response is possible or if the resulting new configuration fails to have the two pebbles in *atom equivalent* states (i.e., the new positions are distinguished by at least one atomic proposition, cf. Definition 29 (ii) below). **I** loses during the game if no further round can be played because both pebbles are in states without successors. An infinite run of the game, which continues through an infinite sequence of rounds played according to the above rules, is won by **II**.

We say that **II** has a *winning strategy* in the bisimulation game starting from configuration $(\mathfrak{M}, w; \mathfrak{M}', w')$, if she has responses to any challenges from the first player that guarantee her to win the game (either because **I** gets stuck, or because she can respond with good moves indefinitely).

Intuitively, we think of **I** as challenging the claim of bisimilarity in the current configuration, while **II** defends that bisimilarity claim. This is borne out by the following.

PROPOSITION 28. *Player* **II** *has a winning strategy in the bisimulation game starting from the initial configuration* $(\mathfrak{M}, w; \mathfrak{M}', w')$ *if, and only if,* $(\mathfrak{M}, w) \rightleftarrows (\mathfrak{M}', w')$.

Indeed, an actual bisimulation $\rho\colon (\mathfrak{M}, w) \rightleftarrows (\mathfrak{M}', w')$ *is* a non-deterministic winning strategy for **II**: she merely needs to select her responses so that the currently pebbled states remain linked by ρ. The atom equivalence condition on ρ guarantees that atom equivalence between pebbled states is maintained; the *forth* condition guarantees a matching response to challenges played by **I** in $\mathfrak{M}$; the *back* condition similarly guarantees a matching response to challenges played in $\mathfrak{M}'$.

Conversely, the set of pairs (u, u') in all configurations $(\mathfrak{M}, u; \mathfrak{M}', u')$ from which **II** has a winning strategy, if non-empty, is a bisimulation.

3.2 Finite bisimulations and characteristic formulae

The games view of a bisimulation suggest that we look at finite approximations corresponding to the existence of winning strategies for a fixed finite number of rounds. These approximations also hold the key to the connection between bisimulation equivalence and modal equivalence. Natural approximations to $\equiv_{\mathrm{ML}}$ are induced by the stratification of ML with respect to the nesting depth of modal formulae (cf. Definition 1), as follows.

DEFINITION 29. For two pointed Kripke τ-structures $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$:

(i) For $n \geq 0$, $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ are ML$_n$*-equivalent*, denoted $(\mathfrak{M}, w) \equiv^n_{\mathrm{ML}} (\mathfrak{M}', w')$, iff they satisfy exactly the same formulae of ML$_n$.

(ii) At the level of $\equiv^0_{\mathrm{ML}}$, we write $w \simeq w'$ instead of $(\mathfrak{M}, w) \equiv^0_{\mathrm{ML}} (\mathfrak{M}', w')$ and say that w and w' are *atom equivalent* (or, isomorphic when viewed as isolated states with atomic propositions according to V, V').

The n-round bisimulation game is played like the (unbounded) bisimulation game but terminates after n rounds (or beforehand if either player loses during one of these rounds). Now **II** also wins if the n-th round is completed without violating atom equivalence. The notion of a winning strategy is correspondingly adapted.

DEFINITION 30. Let $n \geq 0$. Two pointed Kripke structures $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ are

(i) *n-bisimilar*, or *n-bisimulation equivalent*, denoted $(\mathfrak{M}, w) \rightleftarrows_n (\mathfrak{M}', w')$, if **II** has a winning strategy in the n-round bisimulation game starting from $(\mathfrak{M}, w; \mathfrak{M}', w')$.

(ii) *finitely bisimilar*, $(\mathfrak{M}, w) \rightleftarrows_\omega (\mathfrak{M}', w')$, if $(\mathfrak{M}, w) \rightleftarrows_n (\mathfrak{M}', w')$ for all $n \in \mathbb{N}$.

Note that 0-bisimulation equivalence *is* atom equivalence or modal equivalence $\equiv^0_{\mathrm{ML}}$, indistinguishability at the propositional level.

Clearly n-bisimulation equivalence implies m-bisimulation equivalence for any $m \leq n$; (full) bisimulation equivalence implies finite bisimulation equivalence; and finite bisimulation equivalence implies n-bisimulation equivalence for any n. We shall return to the interesting relationship between finite and full bisimulation equivalence below, in connection with the Hennessy–Milner Theorem (theorem 38 below).

A first connection between $\rightleftarrows_n$ and n-equivalence is made in the following.

LEMMA 31. $(\mathfrak{M}, w) \rightleftarrows_n (\mathfrak{M}', w') \;\Rightarrow\; (\mathfrak{M}, w) \equiv^n_{\mathrm{ML}} (\mathfrak{M}', w')$.

Indeed, if $\mathfrak{M}, w \models \varphi$ and $\mathfrak{M}', w' \models \neg\varphi$ for some $\varphi \in \mathrm{ML}_n$, then **I** has a winning strategy in the n-round game from $(\mathfrak{M}, w; \mathfrak{M}', w')$. This is shown by induction on the nesting depth n of the distinguishing formula φ. At level $n = 0$, a distinction in ML_0 means atomic inequivalence – corresponding to a configuration in which **II** has lost.

In the induction step, assume that $(\mathfrak{M}, w)$ is distinguished from $(\mathfrak{M}', w')$ by a formula $\varphi \in \mathrm{ML}_{n+1}$. Propositional connectives in φ can be unravelled so that without loss of generality φ is of the form $\Diamond\psi$ for some $\psi \in \mathrm{ML}_n$. Suppose then that for instance $\mathfrak{M}', w' \models \neg\varphi$, while $\mathfrak{M}, w \models \varphi$. Let in that case **I** move the pebble in $\mathfrak{M}$ from w to some u, where $\mathfrak{M}, u \models \psi$. As $\mathfrak{M}', w' \models \neg\Diamond\psi$, any available response for **II** can only lead to a configuration $(\mathfrak{M}, u; \mathfrak{M}', u')$ in which $(\mathfrak{M}, u)$ and $(\mathfrak{M}', u')$ are distinguished by $\psi \in \mathrm{ML}_n$. Therefore, by the inductive hypothesis, **I** has a winning strategy for the remaining n rounds of the game.

Characteristic formulae

For the converse to the previous lemma, or for capturing the bounded bisimulation game in terms of modal logic, it is essential that the underlying vocabulary is finite: both τ and Φ need to be finite. We again stick with the case of a single binary accessibility relation R, but that restriction is purely for expository simplicity.

The crucial step in the transition from the bisimulation game to modal logic is the formalisation, as a formula $\chi^n_{[\mathfrak{M},w]} \in \mathrm{ML}_n$, of

"**II** has a winning strategy in the n-round game from $(\mathfrak{M}, w; \mathfrak{M}', w')$"

as a property of $(\mathfrak{M}', w')$, for fixed reference structure $(\mathfrak{M}, w)$ and depth n. In fact $\chi^n_{[\mathfrak{M},w]}$ may be constructed by induction on n, simultaneously for all $(\mathfrak{M}, w)$. Along with the induction one observes that $\rightleftarrows_n$ has finite index, and that, correspondingly, we generate only finitely many non-equivalent formulae $\chi^n_{[\mathfrak{M},w]}$ at level n (for finite τ and Φ!).

For $n = 0$, $\chi^0_{[\mathfrak{M},w]}$ is purely propositional and consists of the conjunction of all $p \in \Phi$ that are true in w and all $\neg p$ for those that are false at w. This fixes the atomic equivalence type, as it should.

Inductively, let

$$\chi^{n+1}_{[\mathfrak{M},w]} := \chi^0_{[\mathfrak{M},w]} \wedge \underbrace{\bigwedge_{(w,u)\in R} \Diamond\chi^n_{[\mathfrak{M},u]}}_{\textit{forth}} \quad\wedge\quad \underbrace{\Box \bigvee_{(w,u)\in R} \chi^n_{[\mathfrak{M},u]}}_{\textit{back}} .$$

Even for infinitely branching $\mathfrak{M}$, the conjunctions and disjunctions in this formula remain finite up to logical equivalence as there are only finitely many formulae of the respective kind.

Clearly $\mathfrak{M}, w \models \chi^{n+1}_{[\mathfrak{M},w]}$. But for arbitrary $(\mathfrak{M}', w')$, $\mathfrak{M}', w' \models \chi^{n+1}_{[\mathfrak{M},w]}$ indeed guarantees **II** a wining strategy in the $(n+1)$-round game from $(\mathfrak{M}, w; \mathfrak{M}', w')$. The conjunct $\chi^{0}_{[\mathfrak{M},w]}$ guarantees that the game is not lost already. The *back*-and-*forth* attributions in the two main conjuncts suggest how these are used to guarantee suitable responses, in the first round, to challenges from **I** played in either $\mathfrak{M}$ (*forth*) or $\mathfrak{M}'$ (*back*), respectively.

The *forth* part says that for all moves from w to some u in $\mathfrak{M}$, $\mathfrak{M}', w' \models \Diamond\chi^{n}_{[\mathfrak{M},u]}$, and any R'-successor u' of w' such that $\mathfrak{M}', u' \models \chi^{n}_{[\mathfrak{M},u]}$ provides a response for **II** that will allow her to succeed through another n rounds.

Similarly the *back* part says that for all moves from w' to some u' in $\mathfrak{M}$, $\mathfrak{M}', u' \models \chi^{n}_{[\mathfrak{M},u]}$ for some R-successor u of w in $\mathfrak{M}$ – a response that is good for another n rounds for **II**.

That failure of $\mathfrak{M}', w'$ to satisfy $\chi^{n}_{[\mathfrak{M},w]}$ affords **I** a win within n rounds follows from Lemma 31. Together, these observations yield the following tight connection between the bisimulation game and modal equivalence.

THEOREM 32. *Let* $(\mathfrak{M}, w)$ *and* $(\mathfrak{M}', w')$ *be pointed Kripke structures of the same finite type with finitely many atomic propositions. Then the following are equivalent:*

(i) $(\mathfrak{M}, w) \rightleftarrows_n (\mathfrak{M}', w')$.

(ii) **II** *has a winning strategy in the n-round game from* $(\mathfrak{M}, w; \mathfrak{M}', w')$.

(iii) $\mathfrak{M}', w' \models \chi^{n}_{[\mathfrak{M},w]}$.

(iv) $(\mathfrak{M}, w) \equiv^{n}_{\mathrm{ML}} (\mathfrak{M}', w')$.

As corollaries we obtain a corresponding characterisation of full modal equivalence, and a normal form for ML formulae.

COROLLARY 33. *Over Kripke structures of finite type and with finitely many atomic propositions, finite bisimulation equivalence* $\rightleftarrows_\omega$ *coincides with modal equivalence.*

COROLLARY 34. *Any formula* $\varphi \in \mathrm{ML}_n$ *is logically equivalent to the disjunction* $\bigvee_{\mathfrak{M},w\models\varphi} \chi^{n}_{[\mathfrak{M},w]}$*, which is in fact finite as there are only finitely many such* χ^n *up to logical equivalence (in the vocabulary of* φ*).*

Similarly, for finite vocabularies, any class $\mathcal{C}$ *of pointed Kripke structures that is closed under n-bisimulation is definable in* ML_n *by the disjunction* $\bigvee_{(\mathfrak{M},w)\in\mathcal{C}} \chi^{n}_{[\mathfrak{M},w]}$.

Bisimulation-invariance of ML, Theorem 14, also becomes a simple corollary of the analysis of the game. Indeed, $\varphi \in \mathrm{ML}_n$ is even invariant under n-bisimulation equivalence $\rightleftarrows_n$, which of course implies invariance under (full) bisimulation.

3.3 Finite model property

A logic $\mathcal{L}$ has the *finite model property* (FMP) iff every satisfiable formula of $\mathcal{L}$ is satisfiable in a finite model, i.e., if satisfiability and satisfiability in finite structures coincide for $\mathcal{L}$.

For specific modal logics (e.g., normal extensions of basic modal logic) the implicit restriction to a prescribed class of admissible frames corresponds to a relativisation of the above criterion to the respective classes of (infinite or finite) admissible models. So

the finite model property for S5 say, states that any formula of ML that is satisfiable over some equivalence frame is also satisfiable over some finite equivalence frame.

The finite model property is a characteristic feature of many modal logics. For any modal logic with a recursive axiomatisation (such that it is recursively enumerable for validity) whose class of admissible finite frames is also recursively enumerable, the finite model property provides a standard method for proving decidability.[4] Here we briefly discuss the general *filtration method* for establishing the finite model property for modal logics. For basic modal logic we illustrate in section 3.3 that it even has a *finite tree model property*: every satisfiable formula has a finite tree model.[5]

Filtration

Filtration is the most widely used method for proving the finite model property in modal logics, particularly those determined by classes of frames with specific properties of the accessibility relation. This method is originally due to McKinsey who first applied an algebraic version of it in modal logic. Filtration was introduced in its present form by Lemmon and Scott [91] and further developed and applied by Segerberg [117]. Gabbay [31] introduced a different version, called *selective filtration*. Later, Fisher and Ladner [29] proved the finite model property of propositional dynamic logic PDL using filtration.

Given a formula φ of a modal logic L and a Kripke structure $\mathfrak{M}$ (of type appropriate for L) satisfying φ, we want to produce a *finite* Kripke structure $\tilde{\mathfrak{M}}$ (of appropriate type) satisfying φ. The method of filtration provides a transformation from models $\mathfrak{M}$ to finite models $\tilde{\mathfrak{M}}$ in a uniform manner with respect to φ and $\mathfrak{M}$. Before outlining the construction let us note that the satisfiability of a modal formula φ in a Kripke structure only depends on the truth of the (finitely many) subformulae of φ across that structure. Therefore, two states in a Kripke structure that satisfy the same subformulae of φ are *indistinguishable* from the viewpoint of φ. Sometimes it is necessary to extend the set of subformulae of φ to a wider but still finite set of formulae, called the *closure of* φ and denoted by $cl(\varphi)$. Thus, $cl(\varphi)$ partitions the model into *finitely many equivalence classes* of states, all states in each class satisfying the same subset of $cl(\varphi)$. The underlying idea of the filtration method is to collapse the infinite model to its finite quotient with respect to the equivalence relation generated by that partition, in a way that preserves the truth of all formulae in $cl(\varphi)$, and hence of φ itself.

The equivalence relation itself can be thought of as coarse-grained approximation to bisimulation equivalence that is specific to the given formula φ. It is meant to preserve φ but needs to do so at a coarser level than bisimulation to be of finite index. Note that n-bisimulation $\rightleftarrows_n$ can also serve as a finite index approximation but, because of its graded nature, does not lend itself to taking quotients in the desired global manner. This is because $\mathfrak{M}, u \rightleftarrows_n \mathfrak{M}, u'$ (i.e., that u and u' are of the same n-bisimulation type) does not imply that the same n-bisimulation types are accessible from u and u'.

Here are the formal details. Take any Kripke structure $\mathfrak{M} = \langle W, R, V \rangle$ and a set

[4] However, not every modal logic with the finite model property is decidable; see for instance $K \times K \times K$ in [35].

[5] In general, the finite model property and the tree model property are independent. While the tree model property can account for the decidability of the basic systems of modal logic and many of its extensions and variations (see section 5), it does not apply to axiomatic extensions which impose specific restrictions on the frames that are incompatible with tree-like structures like symmetry or confluence of the accessibility relation.

of formulae Γ, which is assumed to be closed under subformulae, single negation (i.e., if $\varphi \in \Gamma$ is not a negation itself, then $\neg\varphi \in \Gamma$) and under $\Box/\Diamond$ duality. Define an equivalence relation $\sim_\Gamma$ on W as follows:

$$u \sim_\Gamma w \text{ iff for every } \ \psi \in \Gamma : \mathfrak{M}, u \models \psi \Leftrightarrow \mathfrak{M}, w \models \psi.$$

Let $[w]_\Gamma$ be the equivalence class of w with respect to $\sim_\Gamma$ and $W_\Gamma = \{[w]_\Gamma \mid w \in W\}$. Note that if Γ is finite, then W_Γ is finite, too. Further, the valuation V is collapsed to a valuation V_Γ in W_Γ for all $p \in \Gamma$ canonically: $V_\Gamma(p) = \{[w]_\Gamma \mid w \in V(p)\}$; for all other variables q, V_Γ is defined arbitrarily, e.g., $V_\Gamma(q) = \emptyset$.

Now, we say that a Kripke structure $\tilde{\mathfrak{M}} = \langle W_\Gamma, \tilde{R}, V_\Gamma \rangle$ is a *filtration of M with respect to* Γ if for every $\psi \in \Gamma$ and $w \in W : \mathfrak{M}, w \models \psi$ iff $\tilde{\mathfrak{M}}, [w]_\Gamma \models \psi$. With a slight abuse of terminology, we also say that $\tilde{R}$ is a filtration of R with respect to Γ.

There are two simple conditions on the relation $\tilde{R}$ which guarantee that it is a filtration of R with respect to Γ. They give lower and upper bounds for that relation, respectively:

MIN. For every $u, w \in W$, if uRw, then $[u]_\Gamma \tilde{R} [w]_\Gamma$.

MAX. For every $[u]_\Gamma, [w]_\Gamma \in W_\Gamma$, if $[u]_\Gamma \tilde{R} [w]_\Gamma$, then for every $\Box\psi \in \Gamma$:
if $\mathfrak{M}, u \models \Box\psi$, then $\mathfrak{M}, w \models \psi$.[6]

By induction on ψ one can prove that for every $\tilde{R}$ satisfying these conditions, the structure $\tilde{\mathfrak{M}} = \langle W_\Gamma, \tilde{R}, V_\Gamma \rangle$ is indeed a filtration of $\mathfrak{M}$ with respect to Γ, and this claim is known as the *filtration lemma.* Often, the conditions *MIN* and *MAX* are adopted as the definition of a filtration of R, and the filtration lemma then claims that they imply that $\mathfrak{M}_\Gamma$ has the desired property.

Does every Kripke structure have a filtration with respect to any set of formulae Γ? Yes: converting the implication to equivalence in either of the conditions *MIN* and *MAX* defines a relation that satisfies the other condition, too, and hence renders a filtration:

- the *minimal filtration* $\mathfrak{M}_\Gamma^{\min} = \langle W_\Gamma, R_\Gamma^{\min}, V_\Gamma \rangle$, $R_\Gamma^{\min} = \{([u]_\Gamma, [w]_\Gamma) \mid (u, w) \in R\}$;
- the *maximal filtration* $\mathfrak{M}_\Gamma^{\max} = \langle W_\Gamma, R_\Gamma^{\max}, V_\Gamma \rangle$, where $[u]_\Gamma R_\Gamma^{\max} [w]_\Gamma$ holds iff for every $\Box\psi \in \Gamma$, $\mathfrak{M}, u \models \Box\psi$ implies $\mathfrak{M}, w \models \psi$.

Clearly, every relation $\tilde{R}$ such that $R_\Gamma^{\min} \subseteq \tilde{R} \subseteq R_\Gamma^{\max}$ is a filtration, too.

Now, given a formula φ and a pointed Kripke structure $(\mathfrak{M}, u)$ such that $\mathfrak{M}, u \models \varphi$, applying the filtration construction to $\Gamma = cl(\varphi)$ produces a finite pointed Kripke structure $(\tilde{\mathfrak{M}}, [u]_\Gamma)$ that satisfies φ, whence basic modal logic (modal logic K) has the finite model property.

This method can be refined to establish the finite model property for axiomatic extensions L of K, too, by adjusting the definition of $\tilde{R}$ so as to preserve the desired properties of the original structure $\mathfrak{M}$, such as transitivity, linearity etc., or to impose such properties on the resulting structure $\tilde{\mathfrak{M}}$ and thus to eventually guarantee that it is a model appropriate for the desired modal logic L. Examples of filtrations for a number of important modal and temporal logics, such as $T, K4, S4$, and the logics of various linear orderings can be found in [91, 117, 46]. A more general result, extending a theorem of Lewis [92], is [122, Theorem 2.6.8] stating that every modal logic axiomatised by a finite set of shallow formulae (see Section 8.2) admits filtration.

[6] This condition does not depend on the choice of representatives u and w, as $\psi, \Box\psi \in \Gamma$.

Finite tree model property

Before returning to the relationship between bisimulation and finite bisimulation in the next section, we apply the preservation result of Lemma 31 to an alternative, simple proof of the finite model property for basic modal logic, by establishing a stronger *finite tree model property*.

LEMMA 35 (finite tree model property). *For every $n \in \mathbb{N}$, every pointed Kripke structure of finite relational type is n-bisimilar to a finite tree structure. Consequently, any satisfiable formula of* ML *is satisfied at the root of a finite tree.*

Proof. According to section 2.2 the unfolding $\mathfrak{M}[u]$ of $(\mathfrak{M}, u)$ provides a bisimilar tree structure. As we only need n-bisimulation equivalence, we may cut off $\mathfrak{M}[u]$ at depth n from its root u, to obtain a tree structure $(\mathfrak{M}[u] \restriction U^n(u), u) \rightleftarrows_n (\mathfrak{M}, u)$ whose depth is bounded by n, where $U^n(u)$ stands for the set of nodes at distance up to n from u. This tree structure may still be infinite, due to infinite branching. In that case, however, we may prune successors at every node to retain at most one representative of each $\rightleftarrows_n$ equivalence class. As $\rightleftarrows_n$ has finite index (for finite vocabulary; see section 3.2), the resulting tree structure is finite. ❑

Finite branching, as well as a finite bound on the number of bisimulation types, are obvious for finite $\mathfrak{M}$, but a finite pointed Kripke structure $(\mathfrak{M}, u)$ in which a directed cycle is reachable from u cannot be bisimilar to a finite tree structure. Locally, however, this can be achieved in partial unfoldings.

LEMMA 36. *Let $n \in \mathbb{N}$. Every finite pointed Kripke structure $(\mathfrak{M}, u)$ is bisimilar to a finite pointed structure $(\hat{\mathfrak{M}}, \hat{u})$ whose restriction to depth n from the distinguished node $\hat{u}$ is a tree structure.*

Proof. Let $(\mathfrak{M}[u] \restriction U^n(u), u)$ be as in the proof of the last lemma (now finite). For each leaf node of this structure, take a new disjoint isomorphic copy of $\mathfrak{M}$ itself and identify the leaf node with its bisimilar partner node in that copy of $\mathfrak{M}$. The resulting structure is finite, bisimilar to $(\mathfrak{M}, u)$ and tree-like up to distance n from the distinguished node. ❑

Remarks. Results of this type can be carried much further. For instance, a more involved construction yields finite two-way bisimilar companions which are acyclic in restriction to the n-neighbourhood of any node [106, 107]. Such locally acyclic finite *bisimilar covers* are available also in restriction to various other non-elementary classes of frames, e.g., within the classes of all (finite) rooted frames or finite equivalence frames [17].

It is also interesting to note that finite and bisimilar $(\mathfrak{M}, u)$ and $(\mathfrak{M}', u')$ admit finite bisimilar companions $(\hat{\mathfrak{M}}, \hat{u})$ and $(\hat{\mathfrak{M}}', \hat{u}')$, respectively, whose restrictions to depth n from their distinguished nodes $\hat{u}$ and $\hat{u}'$ are even isomorphic tree structures. For this, we take $(\hat{\mathfrak{M}}, \hat{u})$ and $(\hat{\mathfrak{M}}', \hat{u}')$ as from the proof above and modify them by merely attaching extra isomorphic copies of substructures at nodes in the tree parts so as to achieve equal multiplicities for all bisimulation types at each node in the tree parts. It then follows that the tree parts are isomorphic.

3.4 Finite versus full bisimulation

For the relationship between finite and full bisimulation equivalence, we note that finite bisimulation equivalence can be strictly weaker in structures with infinite branching. A typical example of tree structures with $(\mathfrak{M}, w) \rightleftarrows_\omega (\mathfrak{M}', w')$ but $(\mathfrak{M}, w) \not\rightleftarrows (\mathfrak{M}', w')$ is the following.

EXAMPLE 37. Let $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ be tree Kripke structures with trivial valuations, rooted at w and w', respectively. Let the roots have countably many distinct successors u_i, $i \geq 1$ in $\mathfrak{M}$ and u'_i, $i \geq 0$ in $\mathfrak{M}'$. For $i \geq 1$, we let each of u_i and u'_i be the starting point of a simple finite path of length i. We let the extra node u'_0 in $\mathfrak{M}'$ be the root of a simple infinite path. Then $(\mathfrak{M}, w) \not\rightleftarrows (\mathfrak{M}', w')$: let **I** move in $\mathfrak{M}'$ from w' to u'_0; the second player must move to one of the u_i for $i \geq 1$ in $\mathfrak{M}$; let then **I** lead the play in $\mathfrak{M}'$ along the infinite path: **II** gets stuck and loses in round $i+2$ when the end of the length i path from u_i has been reached. On the other hand, $(\mathfrak{M}, w) \rightleftarrows_n (\mathfrak{M}', w')$ for every $n \in \mathbb{N}$, since any two paths of lengths greater than or equal to n look exactly the same in an n-round game.

However, infinite branching is essential to this phenomenon, as the following shows.

THEOREM 38 (Hennessy–Milner theorem). *Let $\mathfrak{M}$ and $\mathfrak{M}'$ both be finitely branching, i.e., every state in either structure has only finitely many immediate successors.*

Then $(\mathfrak{M}, w) \rightleftarrows_\omega (\mathfrak{M}', w')$ implies $(\mathfrak{M}, w) \rightleftarrows (\mathfrak{M}', w')$. Consequently, over finitely branching Kripke structures, modal equivalence coincides with bisimulation equivalence.

Proof. The argument is best given via the games. We claim that **II** can maintain $(\mathfrak{M}, w) \rightleftarrows_\omega (\mathfrak{M}', w')$ indefinitely – which gives her a winning strategy for the infinite game. For instance, let **I** play in $\mathfrak{M}$ and move the pebble from w to u. Suppose that for all responses u' available to **II** in $\mathfrak{M}'$, $(\mathfrak{M}, u) \not\rightleftarrows_\omega (\mathfrak{M}', u')$. As there are only finitely many choices for u' due to finite branching, we can find a sufficiently large $n \in \mathbb{N}$ such that $(\mathfrak{M}, u) \not\rightleftarrows_n (\mathfrak{M}', u')$ for all u' with $(w', u') \in R'$. But this would imply $(\mathfrak{M}, w) \not\rightleftarrows_{n+1} (\mathfrak{M}', w')$, contradicting the assumption $(\mathfrak{M}, w) \rightleftarrows_\omega (\mathfrak{M}', w')$. ❑

Unlike the Hennessy–Milner theorem, which is rather specific for bisimulation, the following observation rests on arguments from classical model theory, to do with saturation properties, and highlights a more general principle that applies to any finitary versus unbounded game equivalences of the Ehrenfeucht–Fraïssé variety; see for instance [108]. *Saturation* properties refer to the realisation of *types*. We think of a type as the formalisation of the properties of an element, through a set of formulae using constants for parameters from a given structure.

With a first-order language L and a set of parameters $A \subseteq W$ of the universe W of some structure $\mathfrak{M}$, associate the expansion of L_A of L with a constant name for each element of A; the corresponding expansion of $\mathfrak{M}$ is denoted $\mathfrak{M}_A$.

DEFINITION 39. An *element type with parameters in A* (in the first-order language L_A) is a set Σ of L_A-formulae in a single free element variable x. The type Σ is a *type of* $\mathfrak{M}_A$ if it is (finitely) *consistent* with the theory of $\mathfrak{M}_A$ in the sense that $\mathfrak{M}_A \models \exists x \bigwedge \Sigma_0$ for every finite $\Sigma_0 \subseteq \Sigma$. The type Σ is *realised* in $\mathfrak{M}_A$ if $\mathfrak{M}_A, w \models \Sigma$ for some element w.

A structure $\mathfrak{M}$ is ω-saturated if for every finite subset A every type of $\mathfrak{M}_A$ is realised in $\mathfrak{M}_A$.

Interesting properties are often expressible by types rather than by an individual formula. For instance, an R-successor of $w \in \mathfrak{M}$ from which there are arbitrarily long R-paths is described by the type $\Sigma_w := \{Rwx\} \cup \{\mathrm{ST}(\Diamond^n\top; x) \mid n \in \mathbb{N}\}$ with parameter w from $\mathfrak{M}$. Note that this is a type of $\mathfrak{M}$ iff there are arbitrarily long R-paths from $w \in \mathfrak{M}$. This does not imply that $\mathfrak{M}$ itself has a realisation of the type – a successor u of w that simultaneously satisfies all the requirements in Σ_w. By compactness, however, every structure $\mathfrak{M}$ has an ω-saturated elementary extension, cf. [12]. Let $\mathfrak{M}^*$ be such an elementary extension of $\mathfrak{M}$. For $w \in \mathfrak{M}$, Σ_w is a type of $\mathfrak{M}^*$ if it is a type of $\mathfrak{M}$. If Σ_w for $w \in \mathfrak{M}^*$ is a type of $\mathfrak{M}^*$, then there also is some R-successor u of w in $\mathfrak{M}^*$ such that $\mathfrak{M}^*, u \models \Diamond^n\top$ for all n; hence Σ_u will also be a type of $\mathfrak{M}^*$ and repeating the argument inductively we find that $\mathfrak{M}^*$ has an infinite path from w. In ω-saturated models, therefore, any element from which there are arbitrarily long paths, will also have an infinite path. Similar reasoning extends to provide responses for **II** in the infinite game over $\mathfrak{M}^*$ to meet any challenge from **I**, provided she has responses that are good for n rounds, for each n. In other words, playing over ω-saturated structures, **II** has a winning strategy in the infinite game whenever she has, for every n, a winning strategy for the n-round game. The proof is analogous to that given for Proposition 87 below; in the terminology to be introduced there, the class of ω-saturated structures has the *Hennessy–Milner property.*

REMARK 40. As shown in section 6.3, $\rightleftarrows_\omega$ coincides with $\rightleftarrows$ in restriction to ω-saturated structures.

In the bisimulation context weaker forms of saturation suffice, and in that sense the Hennessy–Milner theorem may be regarded as a special case. See section 6.3 for more on (modal) saturation.

Bisimulation and infinitary modal equivalence

Since, over infinitely branching structures, equivalence with respect to ordinary modal logic only reaches up to the level of finite bisimulation equivalence, $\rightleftarrows_\omega$, the question of the actual logical counterpart to full bisimulation equivalence arises. The situation is entirely similar to that in classical first-order logic, where it is clarified by Karp's theorem [82] (see also [68]). While the classical Ehrenfeucht–Fraïssé theorem associates finitary game equivalence (the back-and-forth notion of *finite isomorphism* between structures) with elementary equivalence, full infinitary game equivalence (the back-and-forth notion of *partial isomorphism* between structures) corresponds to equivalence with respect to the *infinitary logic* $\mathrm{L}_{\infty\omega}$ whose syntax allows for disjunctions and conjunctions over arbitrary sets of formulae, [68, 108]. In order to extend modal logic ML to its infinitary variant ML_∞, we put the following additional clause for formula formation. If Ψ is any set of formulae of ML_∞, then $\bigwedge\Psi$ and $\bigvee\Psi$ are formulae of ML_∞. These formulae have an ordinal-valued nesting depth, based on the usual rules for the finitary constructors of ML (see Definition 1) together with the extra stipulation that the nesting depth of an infinitary conjunction or disjunction is the supremum of the nesting depths of the constituent formulae. The semantics of the infinite conjunctions and disjunctions is the natural one; with, e.g., $\mathfrak{M}, w \models \bigvee\Psi$ iff $\mathfrak{M}, w \models \psi$ for some $\psi \in \Psi$.

Completely analogous to the treatment of the finitary game in relation to finitary modal logic, we then get the following. (As a side effect of the availability of infinitary conjunctions and disjunctions, we need not restrict the underlying vocabularies to be

finite.) Comparison with the classical version of Karp's theorem highlights the observation that bisimulation is for modal model theory what partial isomorphism is for classical model theory.

THEOREM 41 (Karp's theorem for modal logic). *Let $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ be Kripke structures of the same type. Then the following are equivalent:*

(i) $(\mathfrak{M}, w) \rightleftarrows (\mathfrak{M}', w')$.

(ii) **II** *has a winning strategy in the infinite bisimulation game from* $(\mathfrak{M}, w; \mathfrak{M}', w')$.

(iii) $(\mathfrak{M}, w) \equiv_{\mathrm{ML}_\infty} (\mathfrak{M}', w')$.

Proof. (i) $\Leftrightarrow$ (ii) is obvious. For (ii) $\Rightarrow$ (iii) compare Lemma 31: similar to there, if $(\mathfrak{M}, w)$ is distinguished from $(\mathfrak{M}', w')$ by a formula of nesting depth α, then one can find a move for **I** which will force a successor configuration in which the positions are distinguished at some nesting depth $\beta < \alpha$. By well-foundedness this gives **I** a winning strategy. For (iii) $\Rightarrow$ (ii) one observes that **II** can maintain ML_∞ equivalence indefinitely. ❑

Remark. Characteristic formulae $\chi^\alpha_{[\mathfrak{M},u]}$ with an ordinal parameter α for their nesting depth, can still be defined inductively in a canonical way. (The analogous infinitary formulae for the infinite first-order Ehrenfeucht–Fraïssé game are known as Scott formulae, see for instance [68].) For the infinite game over infinitely branching $\mathfrak{M}$, a position in which players may have infinitely many non-equivalent choices for a next move, is adequately described by an infinite conjunction $\bigwedge \Diamond\varphi_i$ in conjunction with $\Box \bigvee \varphi_i$, where each φ_i describes the bisimulation type of one potential successor in the game over $\mathfrak{M}$, at a nesting depth level that typically needs to be an infinite ordinal. A sufficiently high nesting depth that can be used uniformly across a given $\mathfrak{M}[u]$ is the least ordinal α such that any two states in $\mathfrak{M}[u]$ that are equivalent at nesting depth α in ML_∞ are equivalent at nesting depth $\alpha + 1$. For this α, equivalence at nesting depth α implies equivalence at any nesting depth, i.e., full ML_∞ equivalence, and hence bisimilarity. (The minimal such α is the *closure ordinal* of the co-inductive definition of the bisimulation relation over $\mathfrak{M}[u]$, also compare section 3.5.)

That a given α has this property for $\mathfrak{M}[u]$ is itself expressible in ML_∞. The defining property of α is equivalent to the assertion that, for all $v \in \mathfrak{M}[u]$, $\mathfrak{M}[u] \models \chi^\alpha_{[\mathfrak{M},v]} \rightarrow \chi^{\alpha+1}_{[\mathfrak{M},v]}$. Let ψ^α be the conjunction of all formulae

$$\Box^n \bigwedge_{v \in \mathfrak{M}[u]} \big(\chi^\alpha_{[\mathfrak{M},v]} \rightarrow \chi^{\alpha+1}_{[\mathfrak{M},v]}\big).$$

Then $\mathfrak{M}, u \models \psi^\alpha$ iff within $\mathfrak{M}[u]$, the ML_∞ type at nesting depth $\alpha+1$ is fully determined by the type at nesting depth α. The conjunction $\chi_{[\mathfrak{M},u]} := \psi^\alpha \wedge \chi^\alpha_{[\mathfrak{M},u]}$ for suitable α, characterises $(\mathfrak{M}, u)$ up to bisimulation, thus providing a canonical characteristic formula in ML_∞.

Unlike the definability assertion of Corollary 34, however, bisimulation closure of a class $\mathcal{C}$ of Kripke structures on its own does not guarantee definability in ML_∞. In the example of Observation 42 below, the relevant disjunction of characteristic formulae would be class-sized, and hence not in ML_∞. However, definability of $\mathcal{C}$ in ML_∞ does follow, for instance, if $\mathcal{C}$ comprises only set-many different bisimulation types (which

is in particular the case for the setting of finite model theory, or in restriction to any other class of bounded cardinality). This is sufficient to ensure that $\mathcal{C}$ is definable by a disjunction over characteristic formulae analogous to Corollary 34.

OBSERVATION 42. Well-foundedness, or the class of all pointed Kripke structures $(\mathfrak{M}, u)$ in which there is no infinite path from u, is not definable in infinitary modal logic ML_∞.

This class is definable by the modal μ-calculus L_μ formula $\mu X.\Box X$ (see section 5.2) and hence in monadic second-order logic MSO.[7] On the other hand, well-foundedness is not even definable in infinitary first-order logic $L_{\infty\omega}$, [95]. We sketch a direct proof of non-definability in ML_∞.

For an ordinal α consider the Kripke structure $\mathfrak{M}_\alpha = \langle \{\beta \mid \beta \leq \alpha\}, R\rangle$ with $R = \{(\beta, \beta') \mid \beta' < \beta \leq \alpha\}$ the inverse of the order relation on these ordinals, and its modification $\mathfrak{M}'_\alpha$ with R replaced by $R' = R \cup \{(\alpha, \alpha)\}$. We show by induction on the ordinal γ that $(\mathfrak{M}'_\alpha, \alpha) \equiv_\gamma (\mathfrak{M}_\alpha, \beta)$ (equivalence in ML_∞ up to nesting depth γ) for all $\alpha \geq \beta \geq \gamma$. It follows that no formula of ML_∞ can separate the well-founded $(\mathfrak{M}_\alpha, \alpha)$ from the non-wellfounded $(\mathfrak{M}'_\alpha, \alpha)$ for all α.

The claim is obvious for $\gamma = 0$; also the limit steps are trivial. For the successor step, from γ to $\gamma + 1$, consider $\alpha \geq \beta \geq \gamma + 1$; it suffices to show that then even $\mathfrak{M}'_\alpha, \alpha \models \Diamond\psi \;\Leftrightarrow \mathfrak{M}_\alpha, \beta \models \Diamond\psi$ for ψ of nesting depth γ. The only non-trivial instance of this assertion is when $\mathfrak{M}'_\alpha, \alpha \models \Diamond\psi$ because $\mathfrak{M}'_\alpha, \alpha \models \psi$. But then $\mathfrak{M}_\alpha, \gamma \models \psi$ by the inductive hypothesis. It follows that $\mathfrak{M}_\alpha, \beta \models \Diamond\psi$ as $\beta \geq \gamma + 1$ implies that $(\beta, \gamma) \in R$.

3.5 Largest bisimulations as greatest fixed points

The union of all bisimulation relations between two given Kripke structures is again a bisimulation relation, and hence a maximal bisimulation in the sense of set inclusion. Such *largest bisimulations* can also be defined co-inductively, and be understood as the greatest fixed-point of suitable monotone operators. Again, and purely for expository purposes, we sketch this approach in the simple case of a single accessibility relation R.

Let $X \subseteq W \times W'$, and let $w \in W$ and $w' \in W'$ be atom equivalent ($w \simeq w'$). Let us say that the pair (w, w') has the *back-and-forth property w.r.t.* X iff player **II** has a single round strategy to lead the bisimulation game from $(\mathfrak{M}, w; \mathfrak{M}', w')$ to a configuration $(\mathfrak{M}, u; \mathfrak{M}', u')$ such that $(u, u') \in X$. (Note that the *back*-and-*forth* conditions for a bisimulation relation say that each of its pairs has the *back*-and-*forth* property w.r.t. the relation itself.)

Consider the following operator F on subsets $X \subseteq W \times W'$:

$$F(X) := \big\{(w, w') \in X \mid (w, w') \text{ has the } \textit{back}\text{-and-}\textit{forth} \text{ property w.r.t. } X\big\}.$$

The operator F is monotone in the sense that $X \subseteq Y \Rightarrow F(X) \subseteq F(Y)$. It therefore has a unique greatest fixed point in restriction to any subset of $W \times W'$. We are interested in the greatest fixed point of F that respects atom equivalence, and therefore consider the restriction F_0 of F to $X_0 := \{(u, u') \in W \times W' \mid u \simeq u'\}$. Let $\rho := \mathrm{gfp}(F_0) \subseteq X_0$ be this greatest fixed point. Being a fixed point of F within X_0, ρ respects atom equivalence;

[7] Note that this is definability in the sense of (local) Kripke structure semantics, albeit in a logic which is itself of a second-order nature, and should not be confused with the modal definability of the class of transitive well-founded frames.

being a fixed point of F, ρ has the *back*-and-*forth* property. So ρ is a bisimulation. As any bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$ must also be a fixed-point of F_0, ρ is the largest such.

REMARK 43. The stages of the evaluation of $\mathrm{gfp}(F_0)$ produce a monotone decreasing ordinal-indexed sequence of subsets $X_\alpha \subseteq W \times W'$ according to

$$\begin{array}{rcll} X_0 & = & \{(u,u') \in W \times W' \mid u \simeq u'\} & \\ X_{\alpha+1} & = & F_0(X_\alpha) & \text{(successor stage)} \\ X_\lambda & = & \bigcap_{\alpha<\lambda} X_\alpha & \text{(limit stage)} \end{array}$$

which is eventually constant with value $\mathrm{gfp}(F_0)$. The least ordinal α such that $X_{\alpha+1} = X_\alpha$ is called the *closure ordinal* of this greatest fixed point evaluation over $\mathfrak{M}$ and $\mathfrak{M}'$. This closure ordinal is bounded by the number of bisimulation types realised in $\mathfrak{M}$ and $\mathfrak{M}'$. For cardinality reasons it is in particular strictly less than the successor cardinal of $|W| + |W'|$.

Over finite Kripke structures in particular, the limit $\mathrm{gfp}(F_0)$ is reached within a number of iterations bounded by $|W| + |W'|$, whence the largest bisimulation is polynomial time computable.

One verifies by induction that, for $n \in \mathbb{N}$, X_n is the subset

$$X_n = \big\{(u,u') \in W \times W' \mid (\mathfrak{M}, u) \rightleftarrows_n (\mathfrak{M}', u')\big\}$$

and correspondingly that

$$X_\omega = \big\{(u,u') \in W \times W' \mid (\mathfrak{M}, u) \rightleftarrows_\omega (\mathfrak{M}', u')\big\}.$$

Closure within $m := |W| + |W'|$ steps for finite Kripke structures, implies that, in restriction to $\mathfrak{M}$ and $\mathfrak{M}'$, m-bisimulation equivalence $\rightleftarrows_m$ and hence equivalence in ML_m coincide with full bisimulation equivalence $\rightleftarrows$ and equivalence in ML. This quantitative analysis provides a direct proof of the Hennessy–Milner theorem (with additional a priori bounds) in the special case of finite (rather than just finitely branching) Kripke structures.

3.6 Bisimulation quotients and canonical representatives

Bisimulation quotients provide canonical minimal bisimilar companions, in which every bisimulation type is realised only once. They thus form succinct representations of the overall bisimulation type of a structure $\mathfrak{M}$. There is an analogy with filtrations (compare section 3.3), but here the quotient is taken with respect to the largest bisimulation within the given structure, rather than with respect to some coarser equivalence induced by some set of modal formulae. Passage to bisimulation quotients is often desirable for complexity reasons, for instance for model checking of bisimulation invariant properties. Bisimulation quotients of finite structures are polynomial time computable, as the largest bisimulation is polynomial time computable over finite structures as a greatest fixed point.

For a Kripke structure $\mathfrak{M} = \langle W, \{R_\alpha\}_{\alpha\in\tau}, V\rangle$, consider the largest bisimulation within $\mathfrak{M}$ itself, $\rho^{\mathfrak{M}} = \{(u,u') \mid (\mathfrak{M}, u) \rightleftarrows (\mathfrak{M}, u')\} \subseteq W^2$, as an equivalence relation on W. Let us write $[u]_\rho$ for the equivalence class of $u \in W$. Note that $\rho^{\mathfrak{M}}$ is a congruence w.r.t. the valuation V (by atom equivalence). Therefore V induces a natural quotient valuation $V/\rho^{\mathfrak{M}}$ on the quotient $W/\rho^{\mathfrak{M}}$. While $\rho^{\mathfrak{M}}$ is not in general a congruence w.r.t. the R_α, clearly $(w,u) \in R_\alpha$ implies that for any $w' \in [w]_\rho$ there is $u' \in [u]_\rho$ such that

$(w', u') \in R_\alpha$ (by the back and forth conditions). A natural quotient interpretation for the R_α over $W/\rho^{\mathfrak{M}}$ therefore is

$$R_\alpha/\rho^{\mathfrak{M}} := \{([w]_\rho, [u]_\rho) \in (W/\rho^{\mathfrak{M}})^2 \mid (w, u) \in R_\alpha^{\mathfrak{M}}\}.$$

DEFINITION 44. The *bisimulation quotient* $\mathfrak{M}/\rho^{\mathfrak{M}}$ is the Kripke structure with universe $W/\rho^{\mathfrak{M}} = \{[u]_\rho \mid u \in W\}$, accessibility relations $R_\alpha/\rho^{\mathfrak{M}}$ and valuation $V/\rho^{\mathfrak{M}}$.

LEMMA 45. *The canonical projection* $\pi\colon W \to W/\rho^{\mathfrak{M}}$ *from* $\mathfrak{M}$ *onto its bisimulation quotient* $\mathfrak{M}/\rho^{\mathfrak{M}}$ *is a surjective bounded morphism.*

$\mathfrak{M}/\rho^{\mathfrak{M}}$ is minimal among all globally bisimilar companion structures of $\mathfrak{M}$, as any other such must also have at least one representative of each bisimulation type realised in $\mathfrak{M}$. Moreover, any global bisimulation between two such quotient structures is uniquely determined by bisimulation types and is necessarily an isomorphism. The analogue for ordinary (rather than global) bisimulation equivalence of pointed Kripke structures $(\mathfrak{M}, u)$ needs to be based on quotients $\mathfrak{M}[u]/\rho^{\mathfrak{M}}$ taken after restriction to the generated substructure rooted at u. The bisimulation quotient associated with a (pointed) Kripke structure thus provides a *canonical representative* of its bisimulation type, 'canonical' in the sense of being uniquely determined up to isomorphism.

COROLLARY 46. *Kripke structures* $\mathfrak{M}$ *and* $\mathfrak{M}'$ *are globally bisimilar iff their bisimulation quotients are isomorphic. Pointed Kripke structures* $(\mathfrak{M}, u)$ *and* $(\mathfrak{M}', u')$ *are bisimilar iff the bisimulation quotients* $(\mathfrak{M}[u]/\rho^{\mathfrak{M}}, [u]_\rho^{\mathfrak{M}})$ *and* $(\mathfrak{M}'[u']/\rho^{\mathfrak{M}'}, [u']_\rho^{\mathfrak{M}'})$ *are isomorphic.*

For other kinds of canonical representatives of the bisimulation type of a pointed Kripke structure we may look to trees. Via tree unfoldings any pointed Kripke structure is bisimilar to a tree structure. In order to associate a companion tree structure which is uniquely determined up to isomorphism, though, one needs to impose conditions on the multiplicities among bisimilar siblings in the tree. For countably branching Kripke structures, for instance, in which every state has at most countably many immediate successors, *ω-branching tree unfoldings* $\vec{\mathfrak{M}}^{\omega}[u]$ may be used. These are defined in complete analogy with ordinary tree unfoldings, cf. Definition 21, but based on the set of all ω-labelled paths rooted at u. An ω-labelled path in $\mathfrak{M}$ is a sequence $\vec{w} = (w_0, \alpha_1, m_1, w_1, \ldots, \alpha_k, m_k, w_k)$, where $\vec{w} = (w_0, \alpha_1, w_1, \ldots, \alpha_k, w_k)$ is a path in $\mathfrak{M}$ in the usual sense, and with labels $m_i \in \mathbb{N}$. Two ω-labelled paths $\vec{w}$, $\vec{w}'$ are linked by an R_α-edge in $\vec{\mathfrak{M}}^{\omega}[u]$ if $\vec{w}'$ is an α-extension of $\vec{w}$: $\vec{w}' = (\vec{w}, \alpha, m, w')$ for some $m \in \mathbb{N}$. Through the ω-labelling, the multiplicity of each bisimulation type in each successor set w.r.t. R_α is countably infinite. It is then easy to see that any two bisimilar ω-branching tree unfoldings of countably branching Kripke structures are isomorphic.

This observation may be extended in a straightforward manner to κ-tree unfoldings $\vec{\mathfrak{M}}^{\kappa}[u]$ based on κ-labelled paths, for any infinite cardinal κ.

COROLLARY 47. *For any infinite cardinal* κ*, and pointed Kripke structures* $(\mathfrak{M}, u)$ *and* $(\mathfrak{M}', u')$ *whose branching degree is bounded by* κ*:* $(\mathfrak{M}, u) \rightleftarrows (\mathfrak{M}', u')$ *if, and only if,* $(\vec{\mathfrak{M}}^{\kappa}[u], u) \simeq (\vec{\mathfrak{M}}'^{\kappa}[u'], u')$.

3.7 *Robinson consistency, local interpolation, and Beth definability*

We illustrate the usefulness of the canonicity property expressed in Corollary 47 with a proof of the following analogue of the Robinson joint consistency property [12] for poly-modal logic.

PROPOSITION 48 (Robinson consistency). *For $i = 1, 2$ let $\tau^{(i)}$ be modal similarity types; $\Phi^{(i)}$ sets of atomic propositions; and $\Gamma^{(i)} \subseteq \mathrm{ML}[\tau^{(i)}, \Phi^{(i)}]$. If $\Gamma^{(1)} \cap \Gamma^{(2)}$ is a complete modal theory (in the local sense), and if both $\Gamma^{(1)}$ and $\Gamma^{(2)}$ are consistent, then $\Gamma = \Gamma^{(1)} \cup \Gamma^{(2)}$ is also consistent.*

Proof. Let $\tau^{(0)} := \tau^{(1)} \cap \tau^{(2)}$, $\Phi^{(0)} := \Phi^{(1)} \cap \Phi^{(2)}$, $\Gamma^{(0)} := \Gamma^{(1)} \cap \Gamma^{(2)}$.

Let $\mathfrak{M}, u \models \Gamma^{(1)}$ and $\mathfrak{N}, v \models \Gamma^{(2)}$. Without loss of generality assume that both structures are ω-saturated, which implies that also their $(\tau^{(0)}, \Phi^{(0)})$ reducts $\mathfrak{M}^{(0)}$ and $\mathfrak{N}^{(0)}$ are ω-saturated. Then $(\mathfrak{M}^{(0)}, u) \equiv_{\mathrm{ML}} (\mathfrak{N}^{(0)}, v)$, as both satisfy the complete theory $\Gamma^{(0)}$. By the Hennessy–Milner property for ω-saturated structures: $(\mathfrak{M}^{(0)}, u) \rightleftarrows (\mathfrak{N}^{(0)}, v)$ (cf. Remark 40).

Let $\kappa \geq |\mathfrak{M}|, |\mathfrak{N}|$ and consider the κ-tree unfoldings $\hat{\mathfrak{M}} := \vec{\mathfrak{M}}^{\kappa}[u]$ and $\hat{\mathfrak{N}} := \vec{\mathfrak{N}}^{\kappa}[v]$. The generated $(\tau^{(0)}, \Phi^{(0)})$-subtrees of $(\hat{\mathfrak{M}}, u)$ and $(\hat{\mathfrak{N}}, v)$ are themselves κ-tree unfoldings of corresponding generated $(\tau^{(0)}, \Phi^{(0)})$-substructures of $\mathfrak{M}$ and $\mathfrak{N}$. So they are isomorphic as $(\tau^{(0)}, \Phi^{(0)})$-trees. We may assume that $(\hat{\mathfrak{M}}, u)$ and $(\hat{\mathfrak{N}}, v)$ intersect precisely in these isomorphic subtrees. Let then $\mathfrak{K} := \hat{\mathfrak{M}} \cup \hat{\mathfrak{N}}$ be their union (note that $u = v$). The component structures $(\hat{\mathfrak{M}}, u)$ and $(\hat{\mathfrak{N}}, v)$ are the generated $(\tau^{(i)}, \Phi^{(i)})$-subtrees for $i = 1$ and $i = 2$, respectively. By bisimulation invariance, $\mathfrak{K}, u \models \Gamma^{(i)}$ for $i = 1, 2$. Therefore Γ is satisfiable. ❑

Consistency properties can usually be directly related to interpolation [12]. Here we obtain the *local interpolation* theorem for poly-modal logic as a corollary. For modal similarity types and sets of atomic propositions as above: let $\models \varphi \to \psi$ be a valid (local) consequence, $\varphi \in \mathrm{ML}[\tau^{(1)}, \Phi^{(1)}]$, $\psi \in \mathrm{ML}[\tau^{(2)}, \Phi^{(2)}]$. We want to show that there is an interpolant $\chi \in \mathrm{ML}[\tau^{(0)}, \Phi^{(0)}]$ (i.e., in the common language):

$$\models (\varphi \to \chi) \wedge (\chi \to \psi).$$

Assume there was no interpolant. One can then find a complete theory $\Gamma^{(0)}$ in the common vocabulary for which both $\Gamma^{(1)} := \Gamma^{(0)} \cup \{\varphi\}$ and $\Gamma^{(2)} := \Gamma^{(0)} \cup \{\neg\psi\}$ are consistent (see below). With the consistency property established above, however, this would show that $\varphi \wedge \neg\psi$ is satisfiable, invalidating the implication $\varphi \to \psi$. Assuming without loss of generality that the common language $\mathrm{ML}[\tau^{(0)}, \Phi^{(0)}]$ is countable, one generates $\Gamma^{(0)}$ inductively as a union of an increasing chain of finite sets $\Gamma^{(0)}_n$. The sets $\Gamma^{(0)}_n$ are inductively augmented towards completion, by adding one formula or its negation at a time, guided by the condition that there be no interpolant χ with $\Gamma^{(0)}_n \models (\varphi \to \chi) \wedge (\chi \to \psi)$.

COROLLARY 49. *Poly-modal logic satisfies the interpolation theorem for local consequence.*

The interpolation property can be relativised to particular modal logics or classes of frames. We mention one general result of this kind. Following [122], a subframe $\mathfrak{F}$ of the direct product $\prod_{i \in I} \mathfrak{F}_i$ (see section 6.2) is said to be a *bisimulation product* of the family of frames $\{\mathfrak{F}_i\}_{i \in I}$ if the canonical projection $\pi_i : \mathfrak{F} \to \mathfrak{F}_i$ is a surjective bounded morphism, for each $i \in I$. The following has been established in [122, Thm 2.5.3].

PROPOSITION 50. *Let $\mathcal{K}$ be an elementary class of frames closed under generated subframes and bisimulation products. Then modal logic over $\mathcal{K}$ has interpolation.*

The interpolation property is intimately related to the *Beth definability property* which links *implicit* with *explicit* definability.

Consider a fixed (poly-)modal language $\mathrm{ML}[\tau,\Phi]$. For any list of propositional variables $\mathbf{q}$ from Φ, we denote by $\mathrm{ML}[\mathbf{q}]$ the sublanguage of $\mathrm{ML}[\tau,\Phi]$ restricted to the propositional variables listed in $\mathbf{q}$. Let $p \in \Phi$ be a propositional variable not in $\mathbf{q}$, and $\Gamma = \Gamma(p,\mathbf{q}) \subseteq \mathrm{ML}[p,\mathbf{q}]$ a modal theory. Intuitively, Γ defines p implicitly if it uniquely determines the valuation of p relative to the rest. Formally, let p' be a propositional variable not occurring in $\Gamma(p,\mathbf{q})$ and $\Gamma' = \Gamma(p',\mathbf{q})$ the result of substituting p' for p throughout Γ. Γ *defines p implicitly* if the following is valid (in the sense of local consequence):

$$\Gamma \cup \Gamma' \vDash p \leftrightarrow p'.$$

On the other hand, p is said to be *explicitly definable* relative to Γ if for some $\varphi(\mathbf{q}) \in \mathrm{ML}[\mathbf{q}]$ (thus, not containing p):

$$\Gamma \vDash p \leftrightarrow \varphi(\mathbf{q}).$$

Such φ is then called an *explicit definition* of p relative to Γ.

Clearly, explicit definability entails implicit definability. *Beth's definability theorem* (proved in the early 1950s for first-order logic) states the converse: *implicit definability entails explicit definability.* A standard proof technique is by reduction to interpolation.

Let $\Gamma(p,\mathbf{q}) \cup \Gamma(p',\mathbf{q}) \vDash p \leftrightarrow p'$. By compactness, $\gamma(p,\mathbf{q}) \wedge \gamma(p',\mathbf{q}) \vDash p \leftrightarrow p'$ for some formula γ from Γ (assuming Γ closed under $\wedge$). This implies the validity of

$$\vDash \big(\gamma(p,\mathbf{q}) \wedge p\big) \to \big(\gamma(p',\mathbf{q}) \to p'\big).$$

Local interpolation yields an interpolant $\varphi \in \mathrm{ML}[\mathbf{q}]$ in the common language and thus not containing p or p', such that both $\vDash (\gamma(p,\mathbf{q}) \wedge p) \to \varphi$ and $\vDash \varphi \to (\gamma(p',\mathbf{q}) \to p')$. Together these two establish that φ explicitly defines p relative to γ and hence relative to Γ. We have thus obtained the following.

COROLLARY 51. *Modal logic satisfies Beth's definability theorem for local consequence.*

The notions of interpolation, implicit and explicit definability, and the Beth definability property admit *global* versions, with respect to the global consequence relation (i.e., with respect to validity in Kripke structures). Beth's definability theorem for global consequence can be proved just like the local one above, by noting that Γ implies ψ globally iff $\Box^*\Gamma \vDash \psi$, where $\Box^*\Gamma = \{\Box^n\gamma \mid n \in \mathbb{N}, \gamma \in \Gamma\}$.

Semantically, global implicit definability means that, in any Kripke structure $\mathfrak{M}$ for $\mathrm{ML}(\mathbf{q})$, there is at most one valuation for p such that the resulting expansion $\mathfrak{M}^p$ satisfies $\Gamma(p,\mathbf{q})$. Thus, in order to show that Γ does *not* define p implicitly it suffices to find two models of $\Gamma(p,\mathbf{q})$ that differ in the valuation of p but are otherwise identical. This is the idea of *Padoa's method* for disproving definability in classical logic.

As for global explicit definability in modal logic, Conradie [13] has shown that it can be characterised semantically as follows. p is explicitly globally definable relative to $\Gamma(p,\mathbf{q})$ iff for every two Kripke structures $\mathfrak{M}_1$ and $\mathfrak{M}_2$ satisfying $\Gamma(p,\mathbf{q})$: $\mathfrak{M}_1 \equiv_{\mathrm{ML}[\mathbf{q}]} \mathfrak{M}_2 \;\Rightarrow\; \mathfrak{M}_1 \equiv_{\mathrm{ML}[p,\mathbf{q}]} \mathfrak{M}_2$. Here $\equiv_{\mathrm{ML}[\mathbf{q}]}$ denotes equivalence in $\mathrm{ML}[\mathbf{q}]$. In fact, $\equiv_{\mathrm{ML}[\mathbf{q}]}$ may be replaced for this condition by the corresponding bisimulation relation $\rightleftarrows_{\mathrm{ML}[\mathbf{q}]}$.

For more on interpolation and Beth definability in modal logic, see Chapter 8 of this handbook, [98, 10, 73], as well as [15] for uniform interpolation in the modal mu-calculus, [122] for results on interpolation in extended modal languages, and [36] for a comprehensive exposition of the state of the art on interpolation and definability.

3.8 Bisimulation-safe modal operators

It is easy to find examples of bisimilar pointed Kripke structures $(\mathfrak{M}, w) \rightleftarrows (\mathfrak{M}', w')$, over the modal similarity type with a single modality associated with an accessibility relation R say, such that the corresponding expansions with new accessibility relations interpreted by the converse relations $R^{-1} := \{(u, v) \mid (v, u) \in R\}$ are not bisimilar. On the other hand, $\rho\colon (\mathfrak{M}, w) \rightleftarrows (\mathfrak{M}', w')$ for pointed (poly-modal) Kripke structures $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ implies that the same ρ also is a bisimulation for the expansions by accessibility relations generated from the R_α by the constructors provided in propositional dynamic logic PDL: union, composition, star, as well as test (compare Lemma 70). Thus, the question arises: which operations on relations are 'safe for bisimulations', i.e., preserve bisimulations which hold for their arguments? This question was raised and analysed by van Benthem. In particular, he answered that question completely for the case of *first-order definable* operations on binary relations (see [130, Section 5.3], also [5, Section 2.7]). The operation $\sim$ of *domain-complementation* is defined as an operation on binary relations according to ${\sim}R := \{(x, x) \mid \neg\exists z Rxz\}$.

THEOREM 52. *A first-order definable operation $O(R_1, \ldots, R_n)$ on binary relations is safe for bisimulation iff it can be constructed from $R_1, \ldots, R_n$ using atomic tests $p?$, unions, compositions and the operation of domain-complementation.*

This characterisation was extended in [131] to operations definable in infinitary languages, by allowing infinite unions, too. Since the star operation or iteration, *, is definable as an infinite union of compositions, this accounts for the bisimulation safety of PDL as stated above. The notion of bisimulation-safety and the results above were further extended by Hollenberg [72].

4 MODAL LOGIC AS A FRAGMENT OF FIRST-ORDER LOGIC

The embedding of modal logics into a fragment of first-order logic via the standard translation makes results and techniques for that fragment directly available to the analysis of the modal logic. In this section we discuss further aspects of the relationship between modal and first-order logic.

4.1 Finite variable fragments of first-order logic

DEFINITION 53. Over a purely relational vocabulary and for $k \geq 1$ let *k-variable first-order logic* be the syntactic fragment $\mathrm{FO}^k \subseteq \mathrm{FO}$ consisting of those FO formulae that only use k distinct variable symbols, say $x_0, \ldots, x_{k-1}$, free or bound.

Gabbay [32] first observed that the standard translation, with thrifty re-use of variables as presented in section 1.3, embeds basic modal logic ML into FO^2, the two-variable fragment of first-order logic. (For polyadic modalities of arities up to m, one similarly gets an embedding into the $(m + 1)$-variable fragment.)

LEMMA 54. *The standard translation based on* $\mathrm{ST}(\_;x_0)$ *and* $\mathrm{ST}(\_;x_1)$ *embeds* ML *into* FO^2.

For instance, for ML with a single unary modality $\Diamond$ associated with the binary accessibility relation R, the standard translation operates with an alternate use of two variables, x_0 and x_1, as in

$$\mathrm{ST}\big(\Diamond\Box\Diamond p; x_0\big) = \exists x_1\big(Rx_0x_1 \wedge \forall x_0\big(Rx_1x_0 \rightarrow \exists x_1\big(Rx_0x_1 \wedge Px_1\big)\big)\big).$$

It should be noted that this re-use of variable symbols is at odds for instance with a prenex formalisation in first-order logic. It has several other benefits, however, to be discussed below. And even though the embedding into the guarded fragment of first-order logic which has emerged more recently (see section 4.3 below) may have greater explanatory power for some characteristic features of modal logics, the straightforward embedding into finite variable fragments has also been put to good use.

Consider the embedding of basic modal logic into FO^2. By results of Scott [116] (valid for FO^2 without equality) and Mortimer [102] (with equality), FO^2 has the finite model property. In fact FO^2 has an exponential bound on small models [59]. Therefore, the finite model property for basic modal logic and decidability for satisfiability may be inferred via the translation into FO^2. The complexity and small model bounds obtained in this way, however, are not optimal.

The fact that ML embeds into a finite-variable fragment also provides upper bounds on its model checking complexity. Consider the so-called combined complexity of checking whether $\mathfrak{M}, w \models \varphi$, with both the finite structure $\mathfrak{M}$ and the formula φ as input. The standard translation of modal logic into FO is itself linear time computable. While the combined model checking complexity for FO over finite relational structures is complete for Pspace, it becomes Ptime for FO^k. Moreover, even for FO^2 and basic modal logic the problem is Ptime-hard. For FO^2 one also obtains a bound of $\mathcal{O}(|\varphi||\mathfrak{M}|)$, linear in both input components.[8] FO^2 thus constitutes a natural syntactic fragment of classical first-order logic which matches the finite model property, the decidability and model checking complexity of basic modal logic. These parallels and their limitations are further discussed in [60, 135, 57].

Remark. At the level of FO^3 and higher, which becomes relevant for instance for polyadic modalities, the target logic FO^k fails to have the finite model property and is just as undecidable for satisfiability as full first-order logic, and also does not have linear time model checking. For many purposes, including satisfiability and model checking, however, natural reductions from polyadic into unary modal logics are available that still make the special status of the two-variable fragment available for polyadic modal logics. See for instance [41].

The k-variable fragments of FO play an interesting role in finite model theory and for algorithmic issues, primarily because they give rise to natural and algorithmically manageable *pebble games*. The k-pebble game is precisely the variant of the classical (first-order) Ehrenfeucht–Fraïssé game associated with the restriction to k variable symbols. Bisimulation games in their turn may be regarded as restrictions of these k-pebble games.

[8] This bound refers to a random access model of computation and a succinct representations of the binary accessibility relations R_α through adjacency lists. The input size for the structure is then linear in the number of states plus the number of accessibility edges.

Just as the ordinary n-round Ehrenfeucht–Fraïssé game captures elementary equivalence up to quantifier rank n [26, 68, 108], and just as the n-round bisimulation game captures modal equivalence in ML_n, so the n-round k-pebble game captures equivalence in FO^k up to quantifier rank n.

In the classical (variable-unconstrained) Ehrenfeucht–Fraïssé game for FO over two relational structures $\mathfrak{A}$ versus $\mathfrak{A}'$, the players, **I** and **II**, mark finite configurations of elements in these structures with matching pebbles. A configuration in the game is specified by two tuples of marked elements $\mathbf{a}$ in $\mathfrak{A}$ and $\mathbf{a}'$ in $\mathfrak{A}'$, denoted $(\mathfrak{A}, \mathbf{a}; \mathfrak{A}', \mathbf{a}')$. In each round, **I** chooses one of the structures, and places another marker on one of the elements of that structure; **II** has to respond by marking an element in the opposite structure. In one round the game thus proceeds from a configuration $(\mathfrak{A}, \mathbf{a}; \mathfrak{A}', \mathbf{a}')$ to some configuration $(\mathfrak{A}, \mathbf{a}, a; \mathfrak{A}', \mathbf{a}', a')$ with newly pebbled elements a and a'. **II** loses as soon as the partial map induced by the correspondence between pebbled elements $f\colon \mathbf{a} \mapsto \mathbf{a}'$ is not a local isomorphism. The existence of a winning strategy for **II** in the n-round game then precisely captures elementary equivalence up to quantifier rank n.

The variant for FO^k is obtained by changing the rules in such a manner that no more than k elements of each structure are ever pebbled simultaneously; the game is restricted to configurations $\mathfrak{A}, \mathbf{a}; \mathfrak{A}', \mathbf{a}'$ with tuples $\mathbf{a}$ and $\mathbf{a}'$ of lengths up to k. In any round starting from a configuration of full length k, **I** first removes one of the pebbles and then repositions that same pebble in its structure, and **II** has to do likewise with the matching pebble in the opposite structure. This game then captures levels of equivalence in FO^k, [25].

It is an obvious consequence of Lemma 54 that equivalence in FO^2 implies equivalence with respect to basic modal logic with unary modalities. However, this may also be inferred directly at the level of the games. One observes that the relevant bisimulation game can be emulated by the 2-pebble game in the sense that

- any challenge available to player **I** in the modal game is also available in the 2-pebble game.
- any responses for **II** that are good for the 2-pebble game are good in the modal game, too.

A move along an R-edge in the bisimulation game is emulated in the two-pebble game by means of a placement of the second pebble in the target node. The formerly active pebble now only plays the auxiliary role to guarantee that the right kind of edge is used in an admissible manner also in the response by **II**. But, clearly a strategy in the two-pebble game guarantees more than just bisimulation equivalence, illustrating the gap in expressive power between modal logic and the two-variable fragment of first-order logic into which it can be embedded.

Consider the expressive power of basic (poly-modal) ML over corresponding Kripke structures $\mathfrak{M} = \langle W, \{R_\alpha\}_{\alpha\in\tau}, \{P_i\}\rangle$. Unlike ML, FO^2 formulae generally define binary predicates over Kripke structures. However, the expressive power of FO is also very limited in this respect. As can be inferred from the 2-pebble game, any FO^2-formula $\varphi(x_0, x_1)$ is logically equivalent to a Boolean combination of quantifier free formulae of FO^2 (atomic formulae, including equality) and FO^2 formulae in a single free variable $\psi(x_i)$, $i = 0, 1$. In other words, the expressive power of FO^2, too, is essentially governed by its expressive power in terms of unary relations (properties of single elements in Kripke structures, state properties in process logics). Comparing the expressive power

of basic modal logic ML with that of FO^2 for defining properties of elements and the discriminating powers of bisimulation versus two-pebble game equivalence, basic modal logic is lacking

(i) relativised quantification along backward R_α-edges.

(ii) quantification relativised by (positive or arbitrary) boolean combinations of accessibility relations (including equality).

(iii) unrelativised, global first-order quantification in one variable.

Corresponding features can be added to basic modal logic, as for instance through extensions via *inverse modalities* (in temporal settings: past modalities) interpreted w.r.t. to the converses $R_\alpha^- = \{(v,u) \mid (u,v) \in R_\alpha\}$; a *global modality* interpreted w.r.t. to the full binary relation $U = W \times W$ over universe W; or other constructors for derived accessibilities.[9] An extension of basic modal logic that provides a minimal set of constructs in the above vein so as to precisely capture the expressive power of FO^2, is provided in [96]. A comparison of the satisfiability problems of these two logics shows that there is no polynomial time translation from FO^2 into its modal counterpart, under suitable complexity assumptions. Furthermore, on certain classes of frames extended modal logics can reach the full expressiveness of first-order logic. The most prominent example is Kamp's result in [80] that the temporal language with Since and Until is expressively complete for all first-order definable connectives on the class of Dedekind complete linear orders. This line of work was further developed by others, including Stavi, Gabbay, Venema, Reynolds. For further details see [32, 34], as well as Chapter 11 of this handbook.

4.2 The van Benthem–Rosen characterisation theorem

The fundamental observation that modal logics are embedded into (fragments of) first-order logic via the standard translation immediately calls for the following question. Given an arbitrary first-order formula (in an appropriate vocabulary of Kripke structures), under which conditions is it equivalently expressible in modal logic? In other words, precisely which first-order properties of pointed Kripke structures are expressible in modal logic? Bisimulation invariance is obviously a necessary condition; van Benthem's Theorem says that it is sufficient as well.

Another point of view is also illuminating. Take bisimulation invariance as the fundamentally important semantic notion. It deserves this status for many non-logical reasons, since it is *the* natural notion of process equivalence (thinking of Kripke structures as transition systems), game equivalence (transition systems for games), knowledge equivalence (Kripke structures for knowledge representation), et cetera. From the perspective of first-order logic, then, one would want to isolate the bisimulation invariant properties because just these conform with the underlying semantic intuition. For instance, a first-order property of transition systems captures a property of processes if, and only if, it does not distinguish between bisimulation equivalent transition systems.

Bisimulation invariance is not a decidable property of first-order formulae, as can be seen through reduction of the satisfiability problem. For $\varphi \in \mathrm{FO}(\tau_\Phi \setminus \{R\})$, the formula $\hat{\varphi}(x) := \varphi \wedge Rxx$ is bisimulation invariant iff φ is unsatisfiable. The syntactic subset consisting of those first-order formulae that happen to be bisimulation invariant is

[9] An example of that is the union (*or*) on program formulae in PDL, which, however, is reducible to plain ML in this context, since, for instance $[\alpha \cup \beta]\varphi \equiv [\alpha]\varphi \wedge [\beta]\varphi$ and $\langle \alpha \cup \beta \rangle\varphi \equiv \langle\alpha\rangle\varphi \vee \langle\beta\rangle\varphi$.

therefore *not* the syntax of a reasonable logic. But the characterisation theorem says that modal logic precisely fills this gap. ML, or its translation into FO, provides decidable syntax for just the bisimulation invariant first-order properties; ML is the first-order logic for bisimulation respecting properties.

THEOREM 55 (van Benthem). *Let $\varphi(x) \in \mathrm{FO}$ be in a vocabulary of Kripke structures. Then the following are equivalent:*

(i) *φ is bisimulation invariant: $(\mathfrak{M}, w) \rightleftarrows (\mathfrak{M}', w')$ implies $\mathfrak{M}, w \models \varphi \Leftrightarrow \mathfrak{M}', w' \models \varphi$.*

(ii) *$\varphi(x)$ is logically equivalent to a formula $\tilde{\varphi} \in \mathrm{ML}$.*

Note that (ii) $\Rightarrow$ (i) is just Theorem 14 again. The crucial point here is *expressive completeness* of ML for *all* bisimulation invariant first-order properties. The core idea for that is to establish the following – which is reminiscent of a compactness property.

LEMMA 56. *If $\varphi(x) \in \mathrm{FO}$ is bisimulation invariant, then it is invariant under n-bisimulation for some $n \in \mathbb{N}$.*

The lemma implies (i) $\Rightarrow$ (ii) in the theorem, as any n-bisimulation invariant property is clearly definable in ML_n. Indeed, by Corollary 34, φ is then equivalent to a disjunction of characteristic formulae for n-bisimulation equivalence classes. For the lemma, we sketch a version of the classical proof and an alternative argument more closely based on the games.

Via classical model theory. Assume to the contrary that φ was not invariant under n-bisimulation for any $n \in \mathbb{N}$, and hence not equivalent to any modal formula. Enumerate all modal formulae of the appropriate type as $(\psi_i)_{i\in\mathbb{N}}$. Successively choose one of ψ_i or $\neg\psi_i$ to obtain a maximally consistent set T of modal formulae consistent with both φ and $\neg\varphi$. By compactness one obtains pointed Kripke structures $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ such that both satisfy T, while $\mathfrak{M}, w \models \varphi$ and $\mathfrak{M}', w' \models \neg\varphi$. As $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ satisfy the same complete modal theory, $(\mathfrak{M}, w) \equiv_{\mathrm{ML}} (\mathfrak{M}', w')$ and therefore $(\mathfrak{M}, w) \rightleftarrows_\omega (\mathfrak{M}', w')$. Passage to ω-saturated (or modally saturated, see section 6.3) elementary extensions of $(\tilde{\mathfrak{M}}, w)$ and $(\tilde{\mathfrak{M}}', w')$ would then give us structures $(\tilde{\mathfrak{M}}, w) \rightleftarrows (\tilde{\mathfrak{M}}', w')$ (cf. Remark 40), which are still distinguished by φ, contradicting bisimulation invariance of φ.

Via games. This alternative proof of the crucial step towards the characterisation theorem admits ramifications that persist where the classical argument fails, in particular in finite model theory. In its present form this argument is based on [105, 107] building on ideas from Rosen's finite model theory version of the characterisation theorem [112], as further discussed below (Theorem 61) and in section 9. The *n-neighbourhood* of an element u in a Kripke structure $\mathfrak{M}$ consists of all elements whose *Gaifman distance* from u is at most n. Here Gaifman distance is graph theoretic distance in the undirected graph induced by the symmetrised accessibility relation. We write $\mathfrak{M} \restriction U^n(u)$ for the induced substructure on the n-neighbourhood of u in $\mathfrak{M}$.

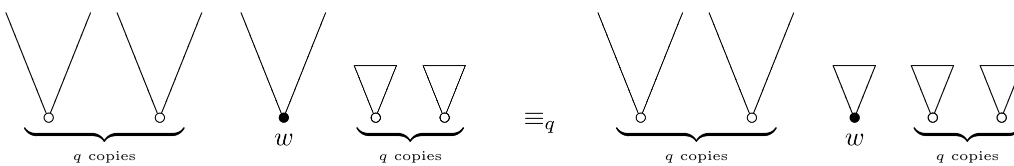
DEFINITION 57. A formula $\varphi(x)$ is *n-local* if for any two pointed tree Kripke structures $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ that are isomorphic in restriction to the n-neighbourhoods of their distinguished nodes, $\mathfrak{M}, w \models \varphi \Leftrightarrow \mathfrak{M}', w' \models \varphi$.

It is easy to see that, if $\varphi(x)$ is bisimulation invariant and n-local, then it is n-bisimulation invariant. In fact, this is obvious for trees and then extends to arbitrary pointed Kripke structures through their unfoldings into trees (see section 2.2). See [105, 106] for the following.

LEMMA 58. *Let $\varphi(x) \in \mathrm{FO}$ have quantifier rank q. If φ is bisimulation invariant, then it is n-local, and hence invariant under n-bisimulation, for $n = 2^q - 1$.*

The first-order locality argument is in fact a ramification of the much more general *Gaifman locality* property of first-order logic [38], which is a useful tool in classical as well as finite model theory [25]. In the context of bisimulation invariant properties, locality together with the exponential bound may however also be derived from a straightforward and self-contained analysis based on first-order Ehrenfeucht–Fraïssé games. In fact, the lemma holds for any $\varphi(x)$ that is invariant under disjoint unions, which itself is an easy consequence of bisimulation invariance (see section 2.2). Let $\varphi(x) \in \mathrm{FO}$ have quantifier rank q. Consider a pointed Kripke structure $(\mathfrak{M}, w)$ or, because it may be conceptually easier though not necessary for the argument, without loss of generality a pointed tree structure $(\mathfrak{M}, w)$ with root w. Let $\mathfrak{M}' = \mathfrak{M} \restriction U^n(w)$ be the substructure induced on the n-neighbourhood of w. It suffices to show that $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M}', w \models \varphi$.

Let $\mathfrak{N}$ be the disjoint union of q copies of $\mathfrak{M}$ and $\mathfrak{M}'$ each. Using invariance under disjoint unions, it suffices to show that $\mathfrak{N} \uplus \mathfrak{M}, w \models \varphi$ iff $\mathfrak{N} \uplus \mathfrak{M}', w \models \varphi$.



It is not hard to exhibit a winning strategy for **II** in the ordinary q-round Ehrenfeucht–Fraïssé game on these structures. **II** merely needs to respect, in round m of the game, the critical distance $d_m = 2^{q-m}$: if **I**'s move in round m goes to within distance d_m of an already pebbled element, **II** plays according to a local isomorphism in the d_m-neighbourhoods of previously pebbled elements; if **I**'s move goes to an element further away from *all* previously pebbled elements, **II** responds in a fresh isomorphic copy of type $\mathfrak{M}$ or $\mathfrak{M}'$, correspondingly.

The exponential bound expressed in the lemma is actually optimal. For a bisimulation invariant property expressible in FO_q but not in ML_n for any $n < 2^q - 1$ consider the property that a state in which p holds is reachable on a path of length less than 2^q. It should be noted that the classical proof of van Benthem's theorem provides no corresponding quantitative information.

COROLLARY 59. *For $\varphi(x) \in \mathrm{FO}$ of quantifier rank q, the following are equivalent for $n = 2^q - 1$:*

(i) *φ is bisimulation invariant.*

(ii) *φ is invariant under n-bisimulation and equivalently expressible in ML_n.*

The exponential bound on the modal nesting depth is sharp: FO *is exponentially more succinct than* ML *for expressing bisimulation invariant properties.*

Some of the underlying ideas of these results are very robust and extend to various ramified settings, some of which are to be discussed in section 5. The classical proof of the characterisation theorem, in particular, carries through for many natural extensions

of basic modal logic associated with refined notions of basic bisimulation equivalence; we mention in particular the corresponding characterisation theorem for the guarded fragment of first-order logic [1] (see Theorem 65 here). But also the game based approach extends to a wide range of settings. One of its main strengths is that it goes through in the setting of finite model theory, as explained below. Another variation that comes naturally from the game based proof is its relativisation to arbitrary bisimulation closed classes [105]. The classical proof, on the other hand, clearly relativises to elementary classes of structures.

COROLLARY 60. *Let $\mathcal{C}$ be a class of Kripke structures that is closed under bisimulation. Then $\varphi(x) \in \mathrm{FO}$ is bisimulation invariant in restriction to $\mathcal{C}$ iff it is equivalent to a formula $\tilde{\varphi} \in \mathrm{ML}$ in restriction to $\mathcal{C}$. Similarly for any elementary class $\mathcal{C}$.*

Theorem 55 characterises the elementary properties of pointed Kripke structures which are definable by single modal formulae. In section 6.4 we will obtain more general preservation results, characterising properties and classes of Kripke structures which are definable by finite or infinite sets of modal formulae, by employing constructions and results from classical model theory.

Ramifications of the characterisation theorem

We sketch a version of the game and locality based proof of van Benthem's characterisation theorem given above, which applies in finite model theory as well as classically. We thus get the finite model theory version due to Rosen [112], even with the same tight exponential bound on succinctness as in Corollary 59.

THEOREM 61. *For $\varphi(x) \in \mathrm{FO}$ of quantifier rank q, the following are equivalent:*

(i) *φ is bisimulation invariant over finite Kripke structures.*

(ii) *φ is equivalent to a formula of ML_n over finite Kripke structures, for $n = 2^q - 1$.*

Proof. We merely need to adapt the proof outlined above in minor ways to avoid passage through infinite structures. For that we may replace bisimilar companion tree structures by the finite, local versions provided by Lemma 36 rather than full unfoldings. For the proof of n-locality of φ (cf. Lemma 58) no modifications are necessary in the game argument, as it applies to arbitrary relational structures exactly as for trees. In fact we only need to use partial tree unfoldings to argue that n-locality and bisimulation invariance together imply n-bisimulation invariance also in restriction to finite structures, as follows.

Let $\varphi(x)$ be bisimulation invariant and n-local over finite structures. Consider finite structures $(\mathfrak{M}, w) \rightleftarrows_n (\mathfrak{M}', w')$. We need to show that $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M}', w' \models \varphi$. As φ is bisimulation invariant, we may replace $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ by bisimilar finite companion structures whose restrictions to $U^n(w)$ and $U^n(w')$ are trees, by Lemma 36. As φ is n-local, these structures may further be replaced by their restrictions to the n-neighbourhoods of w and w', which are n-bisimilar tree structures of depth n, hence bisimilar. So φ is true in w iff it is true in w'. ❑

Compare section 9 for further discussion of the finite model theory context; for further ramifications concerning modal logics based on refined notions of bisimulation also compare section 5; for relativisations to other non-elementary classes of frames, see in particular [17].

4.3 Guarded fragments of first-order logic

The standard translation (see section 1.3) immediately suggested finite variable fragments as an appropriate framework for the study of modal logic within first-order logic. In some ways, however, the finite variables feature fails to give satisfactory insights into the model theoretic behaviour of modal logics. The comparatively smooth finite model theory (see section 9) of modal logics and most notably also their decidability properties (considering robustness under extensions [135]; see section 5.2) are not reflected by the finite variable fragments or even FO^2 in particular [61, 60].

Guarded fragments of first-order logic were introduced by Andréka, van Benthem and Németi in [1]. Compared to the finite variable fragments, the *guarded fragment* GF of first-order logic is much closer to the qualitative characteristics of modal logics. It has greater explanatory power as a framework for the study of modal logics within first-order. For instance, GF and some of its further extensions mirror the decidability as well as finite and tree model properties of modal logics. Crucially, there is a natural notion of *guarded bisimulation* at the root of some of these features. On the other hand guarded logics considerably extend the expressive power of standard modal logics, and in particular still encompass many of their important extensions. Guarded logics have thus come to play an important role in the quest for more expressive fragments of first-order logic that share many of the model theoretic and algorithmic properties that make modal logics so useful for various applications. GF and its relatives extend the scope of essentially modal model theory, including algorithmic and finite model theory aspects, in the direction of first-order.

The guarded fragment GF of FO generalises the *relativised nature* of modal quantification. Let $\alpha(\mathbf{x},\mathbf{y})$ be an atomic first-order formula in variable tuples as displayed, and consider existential and universal quantification over variables $\mathbf{y}$ where the range of quantification is restricted to those $\mathbf{y}$ that satisfy $\alpha(\mathbf{x},\mathbf{y})$ in relation to $\mathbf{x}$ (α is called the *guard* of the quantification). The following shorthand syntax is useful for this α-relativised quantification:

$$(\exists \mathbf{y}.\alpha)\varphi \;:=\; \exists \mathbf{y}\big(\alpha(\mathbf{x},\mathbf{y}) \wedge \varphi(\mathbf{x},\mathbf{y})\big), \text{ and its dual } (\forall \mathbf{y}.\alpha)\varphi \;:=\; \forall \mathbf{y}\big(\alpha(\mathbf{x},\mathbf{y}) \rightarrow \varphi(\mathbf{x},\mathbf{y})\big).$$

Modal quantification (or its standard translation into first-order) displays just this kind of relativisation, where the guards are the atoms $R_\alpha xy$ for accessibility relations R_α.[10] GF admits relativised quantification of this kind, for any atom α, provided that the variables that occur in α comprise *all* the free variables in the formula φ that is being quantified. The standard translation of modal logics (section 1.3) clearly obeys these restrictions.

DEFINITION 62. For an arbitrary relational vocabulary τ, the formulae of $\mathrm{GF}(\tau) \subset \mathrm{FO}(\tau)$, the guarded fragment, are generated from the atomic formulae by closure under boolean connectives and *guarded quantification*; i.e., if $\varphi(\mathbf{x},\mathbf{y}) \in \mathrm{GF}(\tau)$ and if $\alpha(\mathbf{x},\mathbf{y})$ is a τ-atom (also allowing equality) such that $\mathrm{free}(\varphi) \subseteq \mathrm{var}(\alpha)$, then $\big(\forall \mathbf{y}.\alpha\big)\varphi(\mathbf{x},\mathbf{y})$ and $\big(\exists \mathbf{y}.\alpha\big)\varphi(\mathbf{x},\mathbf{y})$ are also in $\mathrm{GF}(\tau)$.

The atom α in these last formulae is called the *guard* of the (universal or existential) quantification. The *nesting depth* is declared for formulae of GF similar to the first-order

[10] This is good also in the polyadic case, where an n-ary modality α associated with an $(n+1)$-ary relation R_α gives rise to quantification with guard $R_\alpha(x,\mathbf{y})$.

quantifier rank, with the only exception that it increases by just 1 with every guarded quantification (rather than by the number of quantified variables in $\mathbf{y}$). The semantics of GF is just that of first-order logic. It makes sense, however, to look at the crucial restriction with a view to a semantic understanding.

DEFINITION 63. Let $\mathfrak{A}$ be a τ-structure. A subset $s \subseteq A$ is *guarded* if s is a singleton set or if $s = \{a_1, \ldots, a_k\}$ for some tuple $(a_1, \ldots, a_k) \in R^{\mathfrak{A}}$ for some relation $R \in \tau$. A tuple $\mathbf{a}$ over $\mathfrak{A}$ is *guarded* if its components are elements of some common guarded subset.

Guarded quantification essentially is quantification over guarded tuples. Intuitively, only the elements of guarded subsets are simultaneously visible in the guarded perspective; this intuition is borne out in the concept of guarded bisimulation (see Definition 64 below).

Clearly the standard translation embeds ML into GF, and actually into the two-variable fragment of GF, $\mathrm{GF} \cap \mathrm{FO}^2$, which is strictly between ML and FO^2 in expressive power, comprising some but not all the features that separate ML from FO^2 as discussed at the end of section 4.1 above. GF naturally comprises

(i) inverse (or past) modalities, as guardedness is non-directional.

(ii) positive Boolean operations on accessibilities (including equality), as for instance in $[\alpha \cap \beta]\varphi \equiv \big(\forall y.(R_\alpha xy \wedge R_\beta xy)\big)\varphi(y) \equiv (\forall y.R_\alpha xy)(R_\beta xy \rightarrow \varphi(y))$.

(iii) a global modality, or universal/existential quantification over a single free variable, as any singleton set is guarded.

Moreover, it should be noted that GF is genuinely polyadic in the sense of representing no restriction on the arities of definable predicates, whereas even polyadic modal logics are still monadic in that sense. But GF indirectly also has a finite variable nature to it. Note that guarded sets are bounded in size by the width (maximal arity) of the available relation symbols. It is not hard to show that any formula in $\mathrm{GF}(\tau)$, for τ of width k, is equivalent to a boolean combination of atomic formulae and formulae that are in $\mathrm{GF} \cap \mathrm{FO}^k$ (up to a possible renaming of variables).

Guarded bisimulations form the backbone of the model theory of GF, playing the same role for GF that ordinary bisimulations play for modal logics. In essence a guarded bisimulation is a back-and-forth equivalence based on local isomorphisms between guarded subsets.

DEFINITION 64. A *guarded bisimulation* between τ-structures $\mathfrak{A}$ and $\mathfrak{B}$ is a non-empty set Z of local (partial) isomorphisms between $\mathfrak{A}$ and $\mathfrak{B}$ such that

(i) for every $\rho \in Z$, the domain and image of ρ are guarded subsets of $\mathfrak{A}$ and $\mathfrak{B}$, respectively.

(ii) Z satisfies the following back-and-forth conditions w.r.t. guarded subsets:
forth: for every $\rho \in Z$ with domain s and every guarded subset s' of $\mathfrak{A}$, there is some $\rho' \in Z$ with domain s' such that ρ and ρ' agree on $s \cap s'$.
back: analogously, w.r.t. to the inverse maps ρ^{-1} and for guarded subsets of $\mathfrak{B}$.

Guarded bisimulations preserve the semantics of GF just as bisimulations preserve the semantics of ML. Moreover, bounded guarded bisimulations – best defined in terms of the restriction of corresponding guarded bisimulation games to a fixed finite number of rounds – precisely capture the levels of equivalence w.r.t. guarded formulae of corresponding nesting depth. Finally, GF is semantically characterised as a fragment of

FO precisely through guarded bisimulation invariance. This analogue of van Benthem's characterisation of modal logic is due to Andréka, van Benthem and Németi [1].

THEOREM 65. *For any first-order formula* $\varphi \in \mathrm{FO}(\tau)$ *the following are equivalent:*

(i) φ *is invariant under guarded bisimulation.*

(ii) φ *is logically equivalent to a formula* $\tilde{\varphi} \in \mathrm{GF}(\tau)$*.*

It is interesting to note that the full analogue of this characterisation theorem in finite model theory is currently still open. For relational vocabularies of width up to two (essentially coloured directed graphs), the analogue is proved in [106].

Perhaps the most important model theoretic consequence of an analysis of GF w.r.t. guarded bisimulations is a corresponding generalisation of the tree model property. For arbitrary relational structures one obtains guarded bisimilar companion structures through a process of *guarded unravelling* or *unfolding*. These relational structures are close to trees in being *tree-decomposable* by means of guarded subsets. Tree decompositions provide a representation of the underlying relational structure by a tree. This notion from graph and hypergraph theory (see for instance [4]) has been fruitfully employed in relational structures also in applications to relational databases [3]. Tree representations based on guarded subsets work with tree structures whose nodes describe all the guarded substructures of the given structure. Guarded unravellings [56, 58] provide tree decompositions by guarded subsets. As the size of guarded subsets in τ-structures is bounded by the width of τ (the maximal arity of relations in τ), one automatically obtains a bound on the tree width. The resulting *generalised tree model property* from [56] is the following.

THEOREM 66. *Any satisfiable formula* $\varphi \in \mathrm{GF}(\tau)$ *has a model which is tree decomposable in terms of its guarded subsets and consequently of tree width* $m-1$*, where* m *is the width of* τ*.*

Such a generalised tree model property can be of eminent model theoretic importance, especially with a view to algorithmic questions, because properties of tree decomposed models may be determined in terms of their tree representations. Using classical model theoretic tools for trees, and in particular automata theoretic methods, the generalised tree model property has strong consequences for decidability and complexity issues. For instance, GF and some of its extensions beyond first-order logic that are invariant under guarded bisimulation and hence satisfy the generalised tree model property, can be decided for satisfiability via reductions to the monadic second-order theory of trees (Rabin's theorem). A direct reduction to emptiness problems for suitable tree automata moreover typically yields optimal complexity bounds. Even finite models for formulae of GF can be built from infinite tree-like models, using finite saturation arguments based on Herwig's extension theorem for partial isomorphisms [67], thus providing an elegant proof of the finite model property for GF [56].

THEOREM 67. *Any satisfiable formula of* GF *has a finite model:* GF *has the finite model property.*

The *clique guarded fragment* pushes the basic idea of guarded quantification a bit further by relaxing the notion of guarded subsets. A subset s of a relational structure is *clique guarded* if any pair of elements from the set is guarded (the subset forms a clique in the Gaifman graph). In the clique guarded fragment, quantification is restricted to

clique guarded rather than guarded subsets. The resulting logic naturally embeds the first-order translation of the Until operator of temporal logic:

$$(\varphi \,\mathsf{Until}\, \psi)(x) \equiv \exists y\big(x \leq y \wedge \psi(y) \wedge \forall z\big((x \leq z \wedge z < y) \rightarrow \varphi(z)\big)\big),$$

because the relevant x, y, z triples form cliques w.r.t. comparability under $\leq$. The clique guarded fragment is no longer restricted to finite variables as clique guarded subsets can have any size. (The Until operator, which crucially requires three variables, is expressible in terms of clique guarded triples w.r.t. a binary relation.)

Despite the increase in expressiveness, the clique guarded fragment is still decidable for satisfiability [56] and it also satisfies the finite model property [69, 70] (with links between clique guardedness and extension theorems for partial isomorphisms).

5 VARIATIONS, EXTENSIONS, AND COMPARISONS OF MODAL LOGICS

There is a considerable body of work on ramifications of the familiar classical modal logics. At the level of ordinary semantics in (pointed) Kripke structures or transition systems, many variations and extensions have been proposed. These largely aim at preserving some of the key model theoretic features of basic modal logics while adapting or boosting the expressive power – either for the purposes of a systematic investigation or for the modelling of situations that cannot be captured by the standard modal languages. The many application areas of modal logics contribute to interesting ramifications and continue to trigger new developments. We give but a few examples. Variants of basic modal languages for the purposes of description logics, as treated in depth in Chapter 13 of this handbook, naturally use for instance inverse modalities (for inverse roles) or graded modalities (for number constraints). Various constructors for new modalities based on composite accessibility relations (e.g., relational composition or transitive closures) have long been studied in temporal and process logics (see Chapters 11 and 12 among others). More recently similar extensions have been employed in formalisms developed for the navigation and retrieval of information in data formats like XML (see [100]).

While a more comprehensive concept of a generalised modal model theory may lead to further consolidation of the big picture, we can here only attempt to exemplify some simple model theoretic ideas in this direction. For a tentative framework, let us regard the underlying notion of bisimulation invariance as the key feature of a specifically modal model theory (at the level of Kripke semantics). We may then tentatively explore this theme along two axes: *variations* in the sense of variations of the underlying notion of bisimulation; and *extensions* of expressive power subject to the requirement of invariance w.r.t. the given notion of bisimulation.

For two typical examples of these orthogonal directions consider, on the one hand, the addition of past modalities (backward moves in the bisimulation game), and, on the other hand, the extension by path quantification (as for reachability assertions or unbounded iteration of $\Diamond$).

For this largely informal sketch we limit ourselves to just a few logics that play a prominent role in connection with transition systems and the behaviour of processes. Some of these and many others are treated at much greater depth in other chapters of this handbook, in particular Chapters 11, 12 and 17 of this handbook and several others in Parts 3 and 4. As criteria for the model theoretic character of the logics

under consideration, over and above their expressive power, we look in particular at the corresponding bisimulation games and model theoretic characterisation theorems, at the tree model property and the finite model property, and at satisfiability issues, which are particularly relevant in many applications (compare Chapters 3 and 17 of this handbook).

5.1 Variations through refined notions of bisimulation

A refinement of bisimulation equivalence ought to be matched, on the logic side, by a more expressive logic. We thus encounter extensions of basic modal logic to more expressive fragments of first-order logic, like those considered in sections 4.1 and 4.3.

In terms of the bisimulation game (or the back-and-forth conditions) over Kripke structures with binary accessibility relations one can introduce a variety of additional moves, in order to capture the expressiveness of some natural extensions of basic modal logic, for instance:

- unconstrained moves to arbitrary states (global bisimulation). This corresponds to the addition of a universal modality (or $\forall/\exists$ quantification) to basic ML, which also allows for an explicit transition between global and local semantics (see, e.g., [54, 22]).
- backward moves along edges (two-way bisimulation). This corresponds to the addition of past or inverse modalities to basic ML.
- counting moves, in which the number of available responses is controlled (counting or locally bijective bisimulation). This corresponds to the extension of basic ML by graded or counting modalities (see [21]).

(Also compare [88] for bisimulations for a hierarchy of description logic languages).

In terms of further reaching variations that also involve the format of the underlying structures and game positions, we discussed in section 4.3 guarded bisimulations for arbitrary relational structures – corresponding to guarded rather than ordinary modal quantification and guarded fragments of first-order logic as important intermediaries between modal and first-order logics.

As indicated, these variations typically correspond to natural extensions of ML. These correspondences manifest themselves in terms of

(i) Ehrenfeucht–Fraïssé relationships: equivalence in the extended logic is characterised by the existence of winning strategies for player **II** in the corresponding, refined bisimulation games.

(ii) characterisation theorems in the style of Theorems 55 or 65 that characterise the respective logic as a fragment of first-order logic, in terms of invariance under the refined notion of bisimulation.

For instance, the *global bisimulation game* gives player **I** the option to switch, for an individual round, to moves in which both players are allowed to move the pebbles to any element of the respective structure rather than just along accessibility edges. This is the Ehrenfeucht–Fraïssé game for the extension ML[$\forall$] of basic ML, in which a global modality is available (corresponding to unrestricted universal first-order quantification in the standard translations). Then **II** has a winning strategy for the n-round game on $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ iff $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ satisfy exactly the same formulae in ML[$\forall$] of

quantifier rank up to n. Also the classical proof pattern for the characterisation theorem (compare the classical proof argument for Lemma 56) goes through. This uses compactness and ω-saturated or modally saturated extensions and the analogue of Remark 40, which is good also for this refined bisimulation game. So we have obtained the following.

PROPOSITION 68. *A first-order formula $\varphi(x)$ is invariant under* global bisimulation *iff it is equivalent to a formula of* ML[∀].

This proposition may serve as a representative for a whole family of similar characterisation results for many other variants of basic modal logic. In fact, these game techniques are not at all even restricted to the modal setting. Analogous Ehrenfeucht–Fraïssé and characterisation theorems hold for instance also for the finite variable fragments FO^k in relation to k-pebble game equivalence. Interestingly, as far as the characterisation theorems are concerned, the picture becomes more varied when we shift attention to the finite model theory versions (cf. section 9, in particular Theorem 130).

5.2 *Extensions beyond first-order*

Extensions induced by variations of the underlying notion of bisimulation in the first instance all lead to modal logics of (pointed) Kripke structures that are still fragments of first-order logic. There is the orthogonal direction of extension that adds expressiveness through stronger constructors in the logic while still adhering to invariance under the given notion of bisimulation. These extensions address some of the expressive deficiencies inherent in first-order, in particular its restriction to essentially local properties (in the sense of Gaifman's locality theorem). Major process logics, aimed at formalising dynamic properties of processes in terms of Kripke structures as transition systems, need to express fundamental properties – like reachability or well-foundedness – that are non-local and hence not expressible in FO.

The process logics discussed below specifically aim for the formalisation of properties of programs or processes, based on the modelling of states and state transitions in Kripke structures as *transition systems*: atomic propositions model atomic *state properties*, and accessibility relations between states model atomic state transformers or *atomic programs*. This setting calls for logics of a fundamentally modal nature – especially since the intended processes are captured by transition systems only up to bisimulation equivalence. Bisimilar transition systems describe exactly the same processes in the sense that there is a complete correspondence of possible runs at the level of individual transitions and in terms of mutual step-wise simulation (*bi-simulation*).

We fix a finite similarity type with modalities α corresponding to binary predicates R_α (transition relations for atomic programs α) and a set of atomic propositions p corresponding to unary predicates P interpreted as the set of states satisfying p. The framework of basic modal logic ML provides modalities for the atomic programs α for assertions about the possible results of single-step state transformations. Various additional constructors have been proposed for the formalisation of dynamic, non-local properties, involving for instance unbounded iterations of transitions. We illustrate the examples of PDL, CTL^* and L_μ. For one simple concrete example of a dynamic, non-local property, we consider the following (at a state):

(χ) in any possible future state of the system, there will be a reachable state in that state's future where p holds.

Propositional dynamic logic

Propositional dynamic logic PDL [29] is based on a dual perspective involving both states and transitions as primary objects of its semantics. Correspondingly, PDL distinguishes two kinds of formulae, *state formulae* and *program formulae*. State formulae, like the familiar modal formulae are evaluated at the states of a transition system and thus define unary predicates on the universe; program formulae on the other hand are evaluated on pairs of states and define binary predicates on the state space, i.e., derived transition relations. Here we work with the following definition; for more on PDL see Chapter 12 of this handbook. We use $\varphi, \psi, \ldots$ for state formulae, $\eta, \zeta, \ldots$ for program formulae.

DEFINITION 69. State and program formulae of PDL are generated by mutual induction.
State formulae: the Boolean closure of atomic propositions p, and modal quantification of the form $\langle\eta\rangle\varphi$ and $[\eta]\varphi$ for program formulae η and state formulae φ.
Program formulae: the closure of the atomic program formulae α and of all formulae $\varphi?$ ("test" operator on state formulae φ) under union $(\eta \cup \zeta)$, composition $(\eta;\zeta)$ and star or iteration, (η^*).

The semantics of state formulae is the natural one based on the semantics of the corresponding program formulae that define modalities η in terms of new transition relations R_η. For those, the specific constructors are defined in relational terms: atomic program formulae α refer to the given transition relations R_α; the union operator is set union: $R_{\eta\cup\zeta} = R_\eta \cup R_\zeta$; composition is relational composition: $R_{\eta;\zeta} = R_\eta \circ R_\zeta = \{(u,w) \mid (u,v) \in R_\eta, (v,w) \in R_\zeta \text{ for some } v\}$; the star operation corresponds to the reflexive transitive closure: $R_{\eta^*} = \bigcup_{n\geq 0}(R_\eta)^n$; finally, the test operator defines a loop relation according to $R_{\varphi?} = \{(u,u) \mid \mathfrak{M}, u \models \varphi\}$.

The PDL state formula $\langle\eta^*\rangle\varphi$, for instance, expresses reachability on an η-path of a state that satisfies φ. Note that this is not expressible in FO, even for atomic η and φ. (χ) of the example above is expressible in PDL using $\eta := \bigcup_{\alpha\in\tau}\alpha$, as $\chi = [\eta^*]\langle\eta^*\rangle p$.

We turn to bisimulation invariance. While the standard notion refers to state formulae, the constructors for PDL program formulae also respect bisimulation equivalence, in the sense of bisimulation safety (see section 3.8).

LEMMA 70. *For Kripke structures* $\mathfrak{M}$, *let* $\mathfrak{M}^*$ *denote the expansion with all the accessibility relations defined by* PDL *program formulae. Then any bisimulation* $\rho\colon \mathfrak{M} \rightleftarrows \mathfrak{M}'$ *is also a bisimulation between these expansions,* $\rho\colon\ \mathfrak{M}^* \rightleftarrows \mathfrak{M}'^*$.

Bisimulation invariance for state formulae is then straightforward. In fact it falls out of the inductive proof of the claim of the lemma, which is best understood in terms of the underlying games. Consider the operations of union, composition and star on accessibility operations. For moves along $R_{\eta\cup\zeta} = R_\eta \cup R_\zeta$, the responses of **II** merely need no longer respect η/ζ individually; moves along $R_{\eta;\zeta}$ can be responded to as if they came as individual moves in two consecutive rounds; similarly, a move along an R_{η^*}-edge corresponds to a finite sequence of moves along R_η-edges, which is similarly covered by **II**'s strategy. If, for some state formula φ, $(u,u') \in \rho$ implies that $\mathfrak{M}, u \models \varphi$ iff $\mathfrak{M}', u' \models \varphi$, then it follows that play according to ρ guarantees that (stationary) $R_{\varphi?}$-moves are available in $\mathfrak{M}$ iff they are available in $\mathfrak{M}'$.

COROLLARY 71. *Any state formula of* PDL *is invariant under bisimulation.*

Computation tree logic

For computation tree logic CTL*, the emphasis is on branching time temporal behaviour rather than process algebra. It is customary to study CTL* over transition systems with a single binary transition relation R (corresponding to a single unary modality $\Diamond$) which moreover is required to have no terminal nodes, i.e., we assume $\mathfrak{M} \models \Diamond\top$.

The intuitive idea in CTL* is to associate the runs from a state u of a transitions system $\mathfrak{M}$ with the tree structure $\vec{\mathfrak{M}}[u]$ (the unfolding or tree unravelling, as defined in section 2.2). The infinite branches of the tree $\vec{\mathfrak{M}}[u]$ are the *computation paths* of $\mathfrak{M}$ at u. Besides state formulae, which define properties of states as usual, CTL* has *path formulae* that define properties of such computation paths. Here a *path* at u is an infinite R-path rooted at u in the usual graph theoretic sense; we write $\sigma = u_0, u_1, \dots$ for a path at $u = u_0$.

DEFINITION 72. State and path formulae of CTL* are generated by mutual induction.
State formulae: Boolean closure of atomic propositions p and formulae $\mathsf{E}\gamma$ and $\mathsf{A}\gamma$ for path formulae γ (existential and universal path quantification).
Path formulae: Boolean closure of all state formulae φ and formulae $\mathsf{Next}\,\gamma$ (temporal "next" operator) and $\gamma\,\mathsf{Until}\,\delta$ (temporal until operator) for path formulae γ, δ.

The semantics of atomic propositions (as state formulae) and of the Boolean connectives is the natural one. We just highlight the specific constructors for state and path formulae. The semantics of a state formula φ is given in terms of a state $u \in \mathfrak{M}$, the semantics of path formulae γ, δ in terms of a path $\sigma = u_0, u_1, \dots$ in $\mathfrak{M}$, whose suffixes we denote as in $\sigma^j = u_j, u_{j+1}, \dots$:

$\mathfrak{M}, u \models \mathsf{E}\gamma$ iff there is a path σ at u such that $\mathfrak{M}, \sigma \models \gamma$, similarly for the dual A.

$\mathfrak{M}, \sigma \models \varphi$ iff $\mathfrak{M}, u_0 \models \varphi$.

$\mathfrak{M}, \sigma \models \mathsf{Next}\,\gamma$ iff $\mathfrak{M}, \sigma^1 \models \varphi$.

$\mathfrak{M}, \sigma \models \gamma\,\mathsf{Until}\,\delta$ iff for some $j \geq 0$: $\mathfrak{M}, \sigma^j \models \delta$ and for $0 \leq i < j$, $\mathfrak{M}, \sigma^i \models \gamma$.

Reachability of a state satisfying φ, for instance, becomes expressible as $\mathsf{E}(\top\,\mathsf{Until}\,\varphi)$. The formula $\top\,\mathsf{Until}\,\varphi$ is also abbreviated $\mathsf{F}\,\varphi$, "eventually φ". Using this abbreviation, our sample property (χ) is expressible as $\chi = \neg\,\mathsf{EF}\,\neg\,\mathsf{EF}\,p$.

PROPOSITION 73. *Any state formula of* CTL* *is invariant under bisimulation.*

This is a straightforward consequence of the fact that any bisimulation $\rho\colon \mathfrak{M} \rightleftarrows \mathfrak{M}'$ preserves paths in the sense that for $(u, u') \in \rho$, every path $\sigma = u_0, u_1, \dots$ at $u_0 = u$ in $\mathfrak{M}$ has a bisimilar companion path $\sigma' = u'_0, u'_1, \dots$ at $u'_0 = u'$ in $\mathfrak{M}'$, which is bisimilar in the sense that $(u_i, u'_i) \in \rho$ for all i.

Interestingly, CTL* admits a characterisation as the bisimulation invariant fragment of monadic path logic, that fragment of monadic second-order logic (over trees) in which second-order quantifiers range over paths. In the light of Theorem 76 below, this characterisation also clarifies the relationship between CTL* and the much more expressive modal μ-calculus. The following is due to [101] over arbitrary tree models and to [65] over the binary tree.

THEOREM 74. *State formulae of* CTL* *precisely define those state properties that are bisimulation invariant and definable in monadic path logic.*

Modal μ-calculus

The modal μ-calculus L_μ is a particularly natural and powerful extension of basic modal logic, which encompasses both PDL and CTL*. In many ways it may be regarded as *the* extension of modal logic for the purposes of temporal reasoning about processes and corresponding model checking applications. Its theory is well developed, ranging from more classical model theoretic issues to computational and in particular automata theoretic analysis; see Chapter 12 of this handbook for a thorough treatment. Here, we only very selectively comment on some aspects of L_μ and essentially restrict ourselves to its role as an extension of ML in our bisimulation-oriented perspective on modal model theory.

L_μ is the canonical fixed point extension of basic modal logic. Least (and dually, greatest) fixed points of monotone operators capture natural forms of recursion closely related to inductive (and dually, co-inductive) definitions. In L_μ basic modal logic is augmented by the means to define, as fixed points, the results of recursions based on definable monotone operators.

Consider basic modal logic with free monadic-second order variables $X, Y, \ldots$ (treated like monadic predicate letters or variables for propositions). A formula $\psi = \psi(X)$ is *positive* in X if X only appears within the scope of an even number of negations in ψ. Positivity in X ensures that, for each structure $\mathfrak{M}$ that interprets all the remaining variables, the following operation on the power set $\mathcal{P}(W)$ of the universe W of $\mathfrak{M}$ is monotone (in the sense that $X \subseteq X'$ implies $\psi[X] \subseteq \psi[X']$):

$$\begin{array}{rcl} \psi^{\mathfrak{M}} \colon \mathcal{P}(W) & \longrightarrow & \mathcal{P}(W) \\ X & \longmapsto & \psi^{\mathfrak{M}}[X] := \{w \in W \mid \mathfrak{M}, X, w \models \psi\}. \end{array}$$

This operation therefore has unique $\subseteq$-minimal and -maximal fixed points, the least and greatest fixed points of $\psi(X)$, respectively.

DEFINITION 75. The syntax of L_μ is based on basic modal logic ML with free monadic second-order variables, plus closure under the least and greatest fixed point constructors: if $\psi \in \mathrm{L}_\mu$ is positive in X, then $\mu X.\psi$ and $\nu X.\psi$ are also formulae of L_μ (in which X is bound).

The semantics of formulae $\varphi \in \mathrm{L}_\mu$ is inductively defined in terms of Kripke structures $\mathfrak{M}$ with interpretations for the free second-order variables; $\mathfrak{M}, u \models \mu X.\psi$ (respectively $\nu X.\psi$) if u is in the least (respectively greatest) fixed point of the operator associated with ψ over $\mathfrak{M}$.

The least fixed point $\mu X.\psi(X)$ in $\mathfrak{M}$ is also definable as the limit of stages X^α generated by induction over the ordinal α, where $X^0 = \emptyset$, $X^{\alpha+1} = \psi^{\mathfrak{M}}[X^\alpha]$ for successor steps, and $X^\lambda = \bigcup_{\alpha<\lambda} X^\alpha$ for limits λ. By monotonicity, the sequence of the X^α is increasing. Over each $\mathfrak{M}$ it eventually must become constant for cardinality reasons. Then the least fixed point of $\psi^{\mathfrak{M}}$ is $X^\infty = \bigcup_\alpha X^\alpha = X^\gamma$ for the minimal γ such that $X^{\gamma+1} = X^\gamma$. (This γ is the closure ordinal of the fixed point over $\mathfrak{M}$.)

The L_μ formula $\mu X.\psi(X)$ for $\psi(X) = \varphi \vee \Diamond X$, for instance, expresses reachability of a state satisfying φ. The monotone operator $\psi^{\mathfrak{M}}$ maps $X \subseteq W$ to the union of $\varphi^{\mathfrak{M}}$ with $\Diamond(X)$. Stage X^n consists of those states from which a state satisfying φ is reachable on an R-path of length less than n. The least fixed point is reached within ω stages over any $\mathfrak{M}$, with $X^\infty = X^\omega$ being the set of states satisfying $\langle R^*\rangle\varphi$. Similarly, well-foundedness

of the converse of R, i.e., non-existence of infinite R-paths from a state, is captured by the least fixed point of the operator defined by the formula $\psi(X) = \Box X$.

Our sample property (χ) is expressible as $\chi = \nu Y.\big(\Box Y \wedge \mu X.(p \vee \Diamond X)\big)$.

Least and greatest fixed points as provided in L_μ admit straightforward explicit definitions in monadic second-order logic MSO, and L_μ may be regarded as a fragment of MSO via a corresponding translation. The following theorem of Janin and Walukiewicz [77] characterises L_μ as the bisimulation invariant fragment of MSO. This is entirely similar in spirit to Theorem 55 for basic modal logic at the first-order level. Covering a far more expressive setting, its proof is also entirely different and based on a sophisticated use of tree automata that recognise corresponding classes of tree models.

THEOREM 76 (Janin–Walukiewicz). *For any* MSO *formula* $\varphi = \varphi(x)$ *the following are equivalent:*

(i) *φ is bisimulation invariant.*

(ii) *φ is logically equivalent to a formula of* L_μ.

We note that, in a similar modal spirit, fixed point extensions have been explored under variations of the underlying notion of bisimulation. In particular, the so-called full μ-calculus with inverse modalities, as related to two-way bisimulation, is studied in [136]; guarded fixed point logic μGF, [62], is the natural extension of the guarded fragment GF by fixed points. For the latter, an analogue of the above characterisation theorem has also been obtained, with a stronger fragment of second-order logic, guarded second-order logic, in place of MSO, [58].

Infinitary modal logics

We encountered ML_∞, the extension of basic modal logic ML by conjunctions and disjunctions over arbitrary *sets* of formulae, in section 3.4. Theorem 41 characterises bisimulation equivalence as equivalence in ML_∞. The restriction to set-size (rather than class-size) disjunctions (or unions) is crucial. Remarkably, L_μ (and CTL*) cannot be embedded into ML_∞: the well-foundedness property expressed by $\mu X.\Box X \in \mathrm{L}_\mu$, for instance, is not globally definable in ML_∞ (see Observation 42). In fact, L_μ (or CTL*) and ML_∞ are incomparable in expressive power.

On the other hand, the individual stages in the generation of any modal least or greatest fixed point are globally definable in ML_∞. In the example of $\mu X.\Box X$, the stages X^α are definable by formulae $\varphi_\alpha \in \mathrm{ML}_\infty$ according to $\varphi_0 = \bot$, $\varphi_{\alpha+1} = \Box\varphi_\alpha$ and $\varphi_\lambda = \bigvee_{\alpha<\lambda} \varphi_\alpha$. The reason that the fixed point X^∞ is not ML_∞ definable is that there is no bound on the closure ordinal of this induction. For many natural (restricted) settings, however, ML_∞ is a maximal bisimulation-invariant logic. For the following compare the remark on characteristic formulae below Theorem 41.

OBSERVATION 77. Over any class of structures that intersects only set-many bisimulation equivalence classes, every bisimulation closed state property is definable in ML_∞.

Several extended logics, including PDL as an important fragment of L_μ, also admit direct translations into ML_∞, though. For PDL this is a consequence of the fact that the closure ordinal of the fixed points needed to capture PDL constructs is uniformly bounded by ω. In fact, PDL therefore embeds into that fragment of ML_∞ in which disjunctions and conjunctions over countable, rather than arbitrary, sets of formulae are

admitted, $\mathrm{ML}_{\omega_1} \subset \mathrm{ML}_\infty$. The PDL reachability assertion $\langle \alpha^* \rangle \varphi$, for instance, globally translates into $\bigvee_{n\in\omega} \langle \alpha \rangle^n \varphi$, where $\langle \alpha \rangle^n$ is the n-fold iteration of the diamond operator.

ML_{ω_1} may be studied as a fragment of the corresponding infinitary extension of first-order logic, $L_{\omega_1\omega}$, which itself has a well developed classical model theory [83]. Similar to $L_{\omega_1\omega}$, ML_{ω_1} also admits a complete proof system (including infinitary rules) and even satisfies (Craig and Lyndon type) interpolation theorems. Characterisation, completeness, and preservation theorems for ML_{ω_1} and some of its fragments have been obtained along such lines by Radev [110] and Sturm [119, 120].

5.3 Model theoretic criteria

We briefly discuss three particularly relevant model theoretic properties in the light of some of the variations and extensions mentioned above. These may serve as examples that among others could contribute to a framework for a more comprehensive comparative model theory of modal logics.

Finite model property (FMP). As noted in section 3.3, the basic modal logic itself has the finite model property, as do many of its variations and extensions. The variations of ML discussed in section 5.1 above, by inverse and global modalities, as well as the guarded fragment GF, have the FMP. For the extensions beyond FO the finite model property for L_μ, due to Streett and Emerson [118], implies FMP for all of its sub-logics, like CTL* and PDL.[11] The *full μ-calculus*, L_μ with inverse modalities, on the other hand lacks the FMP [136]. The following counterexample illustrates this. The formula $\nu X.(\langle R \rangle X \wedge \mu Y. \left[R^{-1}\right] Y)$ requires an infinite (forward) R-path along which every node is well-founded w.r.t. R (does not admit an infinite backward R-path). This implies that the infinite path cannot fold back onto itself; the formula therefore only admits infinite models.

Tree model property. Recall that a logic has the tree model property if every satisfiable formula is satisfied in a tree model. Basic modal logic has the (finite) tree model property (cf. Lemma 35). In fact any bisimulation invariant logic has the tree model property, based on the existence of bisimilar tree unfoldings (cf. section 2.2). In this sense the tree model property, more than the finite model property, is a hallmark of modal model theory. Moreover, many important variations, even though no longer invariant under ordinary bisimulations, still retain (variant) tree model properties. This phenomenon carries particularly far in the case of GF (see Theorem 66, which also generalises to any guarded bisimulation invariant logic).

Decidability. Decidability and complexity of the satisfiability problem provides one measure for the comparison of the variations and extensions discussed above. Basic modal logic may be seen to be decidable for a number of distinct reasons, as it were. Firstly, as FO is recursively enumerable for validity, ML is decidable as a fragment of FO that is recursively enumerable for satisfiability due to its finite model property. More specifically, however, the finite (tree) model property for basic modal logic (cf. Lemma 35) may be strengthened by effective bounds on depth and branching degree of the candidate tree models – indeed, a Pspace (or alternating Ptime) procedure for satisfiability can be

[11] The finite model property of many variations and extensions of modal logic, such as PDL and CTL, can be obtained by filtration, see [46]. However, this method does not work for some of the more complex systems such as CTL* and L_μ, where tableau-like and automata-based methods are applied instead.

extracted (cf. Chapter 3 of this handbook). Alternatively, decidability of ML may be attributed to just its tree model property and the fact that its tree models are recognised by tree automata, for which emptiness is decidable (cf. Chapters 3 and 17). In view of the extensions that go beyond FO this second line of reasoning carries much further. Extensions that are 'modal' in the sense of being bisimulation invariant share the tree model property. Allowing for the appropriate variations of bisimulation, this approach covers not only L_μ, but even the full μ-calculus [136] or the fixed point extension of the guarded fragment [62], which fail to have the FMP. See [57, 135] in this connection for a discussion of the robustness of decidability of modal logics, with a focus on tree models and the accompanying automata theoretic techniques; also see Chapter 17 of this handbook. A comparison between FO^2 and ML in relation to their extensions by natural constructs (e.g., counting, path quantification, transitive closures, fixed points) has also highlighted the special status of modal logic in regard to decidability of such extensions: even comparatively weak extensions of FO^2 along these lines are highly undecidable [61].

6 FURTHER MODEL-THEORETIC CONSTRUCTIONS

One of the traditional directions of development for model theory of a given logic is to identify a sufficiently rich collection of constructions on models, preserving truth in the logic, so that the fundamental concepts of logical definability and logical equivalence can be characterised in terms of these constructions.

In section 2 we introduced the basic model-theoretic notions of generated substructures, bounded morphisms and disjoint unions of Kripke structures and frames, and established corresponding preservation results. These constructions, however, are not sufficient for a complete description of the modal definability of properties or modal equivalence of structures. In this section we introduce and study two more advanced constructions: *ultrafilter extensions* and *ultraproducts*. The former, stemming from the Jónsson–Tarski representation theorem for Boolean algebras with operators in [78], was introduced in modal logic by Goldblatt [43, 44] and used for model-theoretic characterisations of modal definability in [51, 126, 28]. See also section 8. The latter comes from first-order logic, as the most characteristic construction preserving first-order validity (see [12]). Since modal logic on Kripke structures is a fragment of first-order logic, it is a natural truth-preserving construction here, too, and features in the model-theoretic characterisations of modal definability in Kripke structures in section 6.4. Later in this section we indicate how ultrafilter extensions and ultraproducts are linked with each other, and how they relate modal equivalence between Kripke structures with bisimulations, through the notion of saturation.

6.1 Ultrafilter extensions

Let $\mathfrak{F} = \langle W, \{R_\alpha\}_{\alpha\in\tau}\rangle$ be a τ-frame and let $\mathbf{U}(W)$ be the set of all ultrafilters over W. For every $w \in W$, $\mathsf{u}[w] = \{X \subseteq W \mid w \in X\}$ is the principal ultrafilter generated by w. Further, for every $X \subseteq W$ we define $\mathsf{u}(X) := \{\mathsf{u} \in \mathbf{U}(W) \mid X \in \mathsf{u}\}$.

For each $\alpha \in \tau$ we define a binary relation $R^{\mathfrak{ue}}_\alpha$ on $\mathbf{U}(W)$ as follows. For $\mathsf{u}, \mathsf{w} \in \mathbf{U}(W)$:

$$\mathsf{u} R^{\mathfrak{ue}}_\alpha \mathsf{w} \text{ iff } \langle R_\alpha\rangle (X) \in \mathsf{u} \text{ for every } X \in \mathsf{w}.$$

In particular, note that for every $\alpha \in \tau$, and $x, y \in W$, $xR_\alpha y$ iff $\mathsf{u}[x]R^{\mathfrak{ue}}_\alpha\mathsf{u}[y]$.

DEFINITION 78. Given a τ-frame $\mathfrak{F} = \langle W, \{R_\alpha\}_{\alpha\in\tau}\rangle$:

(i) The *ultrafilter extension of* $\mathfrak{F}$ is the τ-frame $\mathfrak{ue}(\mathfrak{F}) := \langle \mathbf{U}(W), \{R_\alpha^{\mathfrak{ue}}\}_{\alpha\in\tau}\rangle$.

(ii) For every Kripke τ-structure $\mathfrak{M} = \langle \mathfrak{F}, V\rangle$, the *ultrafilter extension of* $\mathfrak{M}$ is the Kripke τ-structure $\mathfrak{ue}(\mathfrak{M}) := \langle \mathfrak{ue}(\mathfrak{F}), V^{\mathfrak{ue}}\rangle$ where $V^{\mathfrak{ue}}(p) = \mathsf{u}(V(p))$ for each $p \in \Phi$.

Thus, the subframe of $\mathfrak{ue}(\mathfrak{F})$ consisting of the principal ultrafilters on $\mathfrak{F}$ is isomorphic to $\mathfrak{F}$ but in general, it is *not* a generated subframe of $\mathfrak{ue}(\mathfrak{F})$ (see [5, Example 2.58]). However, every finite frame is isomorphic to its ultrafilter extension. For a proof, see e.g. [5, Proposition 2.59].

Here are two concrete examples of ultrafilter extensions from [129]; also compare [129] for a detailed study of ultrafilter extensions and their use in characterising modal definability in some special classes of frames.

- $\mathfrak{ue}(\langle \mathbb{Z}, <\rangle)$, where $\langle \mathbb{Z}, <\rangle$ is the linearly ordered set of integers, comprises an isomorphic copy of $\langle \mathbb{Z}, <\rangle$ represented by the principal ultrafilters, and two infinite clusters of free ultrafilters, one consisting of elements less than all 'standard' integers, and the other of elements greater than all 'standard' integers. All ultrafilters in each cluster are $<^{\mathfrak{ue}}$-related.

- $\mathfrak{ue}(\langle \mathbb{Q}, <\rangle)$, where $\langle \mathbb{Q}, <\rangle$ is the linearly ordered set of rationals, looks similar. It consists of a copy of the rationals, with infinite clusters on each end, but, since every real number can be approximated from either side by a sequence of rationals, it also has for every *real* number a pair of 'infinitesimally' close clusters, one on either side.

LEMMA 79. *For every Kripke τ-structure $\mathfrak{M} = \langle \mathfrak{F}, V\rangle$ and any formula φ of* ML(τ)*: $V^{\mathfrak{ue}}(\varphi) = \mathsf{u}(V(\varphi))$, i.e., $\mathfrak{ue}(\mathfrak{M}), \mathsf{u} \models \varphi$ iff $V(\varphi) \in \mathsf{u}$.*

This lemma shows that the notion of ultrafilter extension is *canonical*: a state, being an ultrafilter, contains precisely the valuations of those formulae which are true at that state.

COROLLARY 80. *For every Kripke τ-structure $\mathfrak{M} = \langle \mathfrak{F}, V\rangle$, $w \in \mathrm{dom}(\mathfrak{F})$, and any formula φ of* ML(τ)*:*

(i) $\mathfrak{M}, w \models \varphi$ *iff* $\mathfrak{ue}(\mathfrak{M}), \mathsf{u}[w] \models \varphi$.

(ii) *If* $\mathfrak{ue}(\mathfrak{M}) \models \varphi$, *then* $\mathfrak{M} \models \varphi$.

(iii) *If* $\mathfrak{ue}(\mathfrak{F}), \mathsf{u}[w] \models \varphi$, *then* $\mathfrak{F}, w \models \varphi$.

(iv) *If* $\mathfrak{ue}(\mathfrak{F}) \models \varphi$, *then* $\mathfrak{F} \models \varphi$.

We say that a class of τ-frames $\mathcal{C}$ *reflects ultrafilter extensions* if a τ-frame $\mathfrak{F}$ belongs to $\mathcal{C}$ whenever $\mathfrak{ue}(\mathfrak{F}) \in \mathcal{C}$. Thus, FR($\Gamma$) reflects ultrafilter extensions for every set of modal formulae Γ.

That the converses of the latter 3 claims above do not hold can be seen from the following example. The modal formulae preserved in ultrafilter extensions will be characterised in Proposition 114.

EXAMPLE 81. By Proposition 114, the Gödel–Löb formula: $\Box(\Box p \to p) \to \Box p$ is not preserved in ultrafilter extensions because it is not canonical (see [75]).

Non-reflection of ultrafilter extensions can be used to prove modal non-definability in frames in cases where the other truth preserving constructions introduced earlier may not

work. Going back to the example at the end of section 2.3: the sentence $\delta = \forall x \exists y (xRy \wedge yRy)$ is not captured by frame validity of any ML formula, despite being preserved under generated subframes, surjective bounded morphisms and disjoint unions, because it does not reflect ultrafilter extensions. Indeed, $\langle \mathbb{N}, < \rangle \not\models_{\mathrm{FO}} \delta$ while $\mathfrak{ue}(\langle \mathbb{N}, < \rangle) \models_{\mathrm{FO}} \delta$ because every free ultrafilter is a maximal element with respect to the quasi-order $<^{\mathfrak{ue}}$ (see [128] or [5, Example 2.58] for details).

6.2 Ultraproducts

The constructions of direct products and ultraproducts of first-order structures can be applied to frames, considered as $\mathrm{FO}(\tau)$-structures, and to Kripke structures, considered as $\mathrm{FO}(\tau_\Phi)$-structures.

DEFINITION 82. Let $\{W^i\}_{i\in I}$ be a family of sets indexed by a set I.

(i) The *direct product of* $\{W^i\}_{i\in I}$ is the set $\prod_{i\in I} W^i = \{g : I \to \bigcup_{i\in I} W^i \mid g(i) \in W^i \text{ for all } i \in I\}$.

(ii) For any ultrafilter $\mathbf{U}$ on I, the *ultraproduct of* $\{W^i\}_{i\in I}$ *over* $\mathbf{U}$, $\prod^{\mathbf{U}}_{i\in I} W^i$, is the quotient of $\prod_{i\in I} W^i$ w.r.t. the equivalence relation $\sim_{\mathbf{U}}$ defined by $g \sim_{\mathbf{U}} g'$ iff $\{i \in I \mid g(i) = g'(i)\} \in \mathbf{U}$. We write $g^{\mathbf{U}}$ for the $\sim_{\mathbf{U}}$ equivalence class of g.

(iii) For any family $\{X^i \subseteq W^i\}_{i\in I}$, $\prod^{\mathbf{U}}_{i\in I} X^i = \{g^{\mathbf{U}} \in \prod^{\mathbf{U}}_{i\in I} W^i \mid \{i \in I \mid g(i) \in X^i\} \in \mathbf{U}\}$.

DEFINITION 83. Let $\{\mathfrak{F}^i = \langle W^i, \{R^i_\alpha\}_{\alpha\in\tau}\rangle\}_{i\in I}$ be a family of τ-frames indexed by a set I, and $\{\mathfrak{M}^i = \langle \mathfrak{F}^i, V^i\rangle\}_{i\in I}$ be a family of Kripke τ-structures over these frames.

(i) The *direct product of* $\{\mathfrak{F}^i\}_{i\in I}$ is the τ-frame $\prod_{i\in I} \mathfrak{F}^i := \langle \prod_{i\in I} W^i, \{R_\alpha\}_{\alpha\in\tau}\rangle$, where for $\alpha \in \tau$: $g_0 R_\alpha g_1$ iff $g_0(i) R^i_\alpha g_1(i)$ for every $i \in I$.

(ii) The *direct product of* $\{\mathfrak{M}^i\}_{i\in I}$ is the Kripke τ-structure $\prod_{i\in I} \mathfrak{M}^i := \langle \prod_{i\in I} \mathfrak{F}^i, V\rangle$, where $V(p) := \prod_{i\in I} V^i(p)$ for each $p \in \Phi$.

If, further, $\mathbf{U}$ is an ultrafilter on I:

(iii) The *ultraproduct of* $\{\mathfrak{F}^i\}_{i\in I}$ *over* $\mathbf{U}$ is the τ-frame $\prod^{\mathbf{U}}_{i\in I} \mathfrak{F}^i := \langle \prod^{\mathbf{U}}_{i\in I} W^i, \{R^{\mathbf{U}}_\alpha\}_{\alpha\in\tau}\rangle$, where for $\alpha \in \tau$: $g_0^{\mathbf{U}} R^{\mathbf{U}}_\alpha g_1^{\mathbf{U}}$ iff $\{i \in I \mid g_0(i) R^i_\alpha g_1(i)\} \in \mathbf{U}$.

(iv) The *ultraproduct of* $\{\mathfrak{M}^i\}_{i\in I}$ *over* $\mathbf{U}$ is the Kripke τ-structure $\prod^{\mathbf{U}}_{i\in I} \mathfrak{M}^i := \langle \prod^{\mathbf{U}}_{i\in I} \mathfrak{F}^i, V^{\mathbf{U}}\rangle$ such that for each $p \in \Phi$, $V^{\mathbf{U}}(p) := \prod^{\mathbf{U}}_{i\in I} V^i(p)$.

If $\mathfrak{F}^i = \mathfrak{F}$ for every $i \in I$, the ultraproduct is called an *ultrapower of* $\mathfrak{F}$, denoted $\prod^{\mathbf{U}}_I \mathfrak{F}$; similarly for Kripke structures, where the ultrapower is denoted $\prod^{\mathbf{U}}_I \mathfrak{M}$.

By the fundamental theorem of Łoś (see, e.g., [12, 68]), every first-order definable property holds in an ultraproduct iff it holds in a 'large' (i.e., in the ultrafilter) set of component structures. Moreover, every Σ^1_1-definable property is preserved by ultraproducts [12, Corollary 4.1.14]. Therefore, validity of modal formulae in (pointed) frames, being a Π^1_1-definable property in terms of the standard translation, is reflected (i.e., its negation is preserved) by ultraproducts. Using these, we obtain the following preservation results.

PROPOSITION 84. *For every family of Kripke τ-structures $\{\mathfrak{M}^i = \langle \mathfrak{F}^i, V^i\rangle\}_{i\in I}$, ultrafilter $\mathbf{U}$ on I, $g^{\mathbf{U}} \in \prod^{\mathbf{U}}_{i\in I} \mathfrak{F}^i$, and formula φ of* $\mathrm{ML}(\tau)$*:*

(i) $\prod_{i\in}^{\mathbf{U}} \mathfrak{M}^i, g^{\mathbf{U}} \models \varphi$ *iff* $\{j \in I \mid \mathfrak{M}^j, g(j) \models \varphi\} \in \mathbf{U}$.

(ii) $\prod_{i\in I}^{\mathbf{U}} \mathfrak{M}^i \models \varphi$ *iff* $\{j \in I \mid \mathfrak{M}^j \models \varphi\} \in \mathbf{U}$.

(iii) *If* $\prod_{i\in I}^{\mathbf{U}} \mathfrak{F}^i, g^{\mathbf{U}} \models \varphi$, *then* $\{j \in I \mid \mathfrak{F}^j, g(j) \models \varphi\} \in \mathbf{U}$.

(iv) *If* $\prod_{i\in I}^{\mathbf{U}} \mathfrak{F}^i \models \varphi$, *then* $\{j \in I \mid \mathfrak{F}^j \models \varphi\} \in \mathbf{U}$.

Since, however, not every valuation in an ultraproduct of frames can be obtained as an ultraproduct of valuations in the components, the converse of the latter two claims above does not hold.

The following observation due to Goldblatt [44, 47] blends first-order and modal constructions.

PROPOSITION 85. *For any family* $\{\mathfrak{F}^i\}_{i\in I}$ *of* τ*-frames and any ultrafilter* $\mathbf{U}$ *on* I, $\prod_{i\in I}^{\mathbf{U}} \mathfrak{F}^i$ *is embeddable as a generated subframe into* $\prod^{\mathbf{U}} \left(\biguplus_{i\in I} \mathfrak{F}^i\right)$.

The embedding is defined canonically as $g^{\mathbf{U}} \mapsto g^{\mathbf{U},+}$, where $g^{\mathbf{U},+} := (w(i), i)$ for each $i \in I$. Furthermore, as shown in [129], any ultraproduct of frames $\prod_{i\in I}^{\mathbf{U}} \mathfrak{F}^i$ is embeddable as a subframe of $\mathfrak{ue}\left(\biguplus_{i\in I} \mathfrak{F}^i\right)$.

6.3 Modal saturation and bisimulations

A class of (pointed) Kripke structures $\mathcal{C}$ is said to have the *Hennessy–Milner property* if modal equivalence between structures in $\mathcal{C}$ implies (and hence is equivalent to) bisimulation equivalence. For instance, as noted in Theorem 38 the class of all finite structures has the Hennessy–Milner property. Compare Definition 39 for first-order types and ω-saturation. The following weaker notion of saturation is more specific to modal logic.

DEFINITION 86. A Kripke τ-structure $\mathfrak{M} = \langle W, \{R_\alpha\}_{\alpha\in\tau}, V\rangle$ is *modally saturated at a state* $w \in W$ if for every $\alpha \in \tau$ and set of modal formulae Γ, the following saturation condition holds:
if $\mathfrak{M}, w \models \langle\alpha\rangle \bigwedge \Gamma_0$ for all finite $\Gamma_0 \subseteq \Gamma$, then there is some $u \in W$ such that $wR_\alpha u$ and $\mathfrak{M}, u \models \Gamma$.
$\mathfrak{M}$ is *modally saturated* if it is modally saturated at each of its states.

It is clear from Definition 39 that ω-saturated Kripke structures are modally saturated.

PROPOSITION 87. *The class of modally saturated Kripke structures has the Hennessy–Milner property.*

Proof. If $\mathfrak{M}$ and $\mathfrak{M}'$ are modally saturated, then $\rho := \{(w, w') \in W \times W' \mid (\mathfrak{M}, w) \equiv_{\mathrm{ML}} (\mathfrak{M}', w')\}$ is a bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$. Atom equivalence is obvious. Consider for instance the *forth* condition. Let $(\mathfrak{M}, w) \equiv_{\mathrm{ML}} (\mathfrak{M}', w')$ and let $(w, u) \in R_\alpha$. Put $\Gamma := \mathrm{Th}_{\mathrm{ML}}(\mathfrak{M}, u)$. For finite $\Gamma_0 \subseteq \Gamma$, $\mathfrak{M}, w \models \langle\alpha\rangle \bigwedge \Gamma_0$ and hence also $\mathfrak{M}', w' \models \langle\alpha\rangle \bigwedge \Gamma_0$. By modal saturation of $\mathfrak{M}'$ at w' therefore, there is some u' such that $(w', u') \in R_\alpha$ and $\mathfrak{M}', u' \models \Gamma$. But this means that $(\mathfrak{M}, u) \equiv_{\mathrm{ML}} (\mathfrak{M}', u')$, and u' is as desired for the forth requirement. ❑

COROLLARY 88. *The class of* ω*-saturated Kripke structures has the Hennessy–Milner property.*

It is well-known from classical model theory [12, Corollary 4.3.14] that the ultrapower of any (pointed) Kripke structure w.r.t. a regular ultrafilter is an ω-saturated elementary

extension of that structure. Furthermore, two (pointed) Kripke structures are modally equivalent iff any pair of their ω-saturated ultrapowers are modally equivalent, and hence, by Corollary 88, bisimilar. Thus, we obtain the following characterisation of modal equivalence between Kripke structures from [20], as a corollary of the above.

THEOREM 89. *Two (pointed) Kripke structures are modally equivalent iff any pair of their ω-saturated ultrapowers are bisimilar.*

A parallel with first-order logic can be drawn here if we think of bisimulations as the modal analogue of partial isomorphisms between Kripke structures, and note that elementary equivalence on ω-saturated structures coincides with partial isomorphism between them (see [108, 68, 23]). Then Theorem 91 below completes the match. Before getting there, we need the following result, due to van Benthem [126], building on a construction of Fine [28].

THEOREM 90. *For every Kripke τ-structure $\mathfrak{M}$, $\mathfrak{ue}(\mathfrak{M})$ is a bounded morphic image of an ω-saturated ultrapower of $\mathfrak{M}$.*

Proof. Let $\mathfrak{M} = \langle \mathfrak{F}, V\rangle$ where $\mathfrak{F} = \langle W, \{R_\alpha\}_{\alpha\in\tau}\rangle$.
The structure $\mathfrak{F}^\times = \langle W, \{R_\alpha\}_{\alpha\in\tau}, \{X \mid X \subseteq W\}\rangle$ has in particular every $V(\varphi)$ as a distinguished predicate. Take an ω-saturated ultrapower $\mathfrak{F}^* = \prod_I^{\mathbf{U}} \mathfrak{F}^\times$ and for each $f^{\mathbf{U}} \in \prod_I^{\mathbf{U}} W$ define $v(f^{\mathbf{U}}) = \{X \subseteq W \mid f^{\mathbf{U}} \in \prod_I^{\mathbf{U}} X\}$. It is immediate to check that $v(f^{\mathbf{U}}) \in \mathbf{U}(W)$. Considering v as a mapping from $\prod_I^{\mathbf{U}} \mathfrak{M}$ onto $\mathfrak{ue}(\mathfrak{M})$ one can show that it is a bounded morphism. The most difficult step (proved in [126] for the case of one unary modality, see also the proof of [5, Proposition 2.61]) is to prove the *back* condition, which uses the saturation of $\mathfrak{F}^\times$. ❑

Using this theorem we can now obtain a strengthening of the model-theoretic characterisation of modal equivalence, first proved by Hollenberg [71]. See also [138] and [5, Theorem 2.62].

THEOREM 91. *For any pointed Kripke structures* $(\mathfrak{M}, w)$ *and* $(\mathfrak{M}', w')$,

$$(\mathfrak{M}, w) \equiv_{\mathrm{ML}} (\mathfrak{M}', w') \;\; \textit{iff} \;\; (\mathfrak{ue}(\mathfrak{M}), \mathsf{u}[w]) \rightleftarrows (\mathfrak{ue}(\mathfrak{M}'), \mathsf{u}[w']).$$

Proof. The direction from right to left is immediate from Lemma 79 and bisimulation invariance, Theorem 14. For the converse direction, suppose $(\mathfrak{M}, w) \equiv_{\mathrm{ML}} (\mathfrak{M}', w')$. Then, by Theorem 89, $(\prod_I^{\mathbf{U}} \mathfrak{M}, g_w^{\mathbf{U}}) \rightleftarrows (\prod_I^{\mathbf{U}} \mathfrak{M}', g_{w'}^{\mathbf{U}})$ for the ω-saturated ultrapowers defined in the proof above, where $g_w(i) = w$ for each $i \in I$, and likewise for $g_{w'}$. Note that $v(g_w^{\mathbf{U}}) = \mathsf{u}[w]$ and $v(g_{w'}^{\mathbf{U}}) = \mathsf{u}[w']$. Composing this bisimulation with the surjective bounded morphisms $v : (\prod_I^{\mathbf{U}} \mathfrak{M}, g_w^{\mathbf{U}}) \xrightarrow{\rightleftarrows} (\mathfrak{ue}(\mathfrak{M}), \mathsf{u}[w])$ and $v' : (\prod_I^{\mathbf{U}} \mathfrak{M}', g_{w'}^{\mathbf{U}}) \xrightarrow{\rightleftarrows} (\mathfrak{ue}(\mathfrak{M}'), \mathsf{u}[w'])$, we obtain a bisimulation between the ultrafilter extensions. ❑

The following observation is immediate from the definitions.

LEMMA 92. *Bisimulations preserve modal saturation at a state: if $(\mathfrak{M}, w) \rightleftarrows (\mathfrak{M}', w')$, then $\mathfrak{M}$ is modally saturated at w iff $\mathfrak{M}'$ is modally saturated at w'. Consequently, global bisimulations preserve modal saturation of models.*

From this lemma and Theorem 90, since surjective bounded morphisms are global bisimulations, we obtain the following result from [48], (see also [5, Proposition 2.61])

COROLLARY 93. *The ultrafilter extension of every Kripke structure is modally saturated.*

As Venema argues quite aptly in [138], this result along with Theorem 91 indicates that, for modal logics, ultrafilter extensions can play the role that ultrapowers play in first-order logic for the construction of saturated extensions of structures.

6.4 Modal definability of properties of Kripke structures

Kripke structures serve to give model theoretic semantics to modal logic. Conversely, focusing on Kripke structures in their own right, we regard modal logic as a language for defining classes of Kripke structures. We may ask the natural model-theoretic questions from this angle, like, for instance: what classes/properties of (pointed) Kripke structures are definable by (sets of) modal formulae? A definitive answer to that question was given in the case of elementary properties of pointed Kripke structures defined by single modal formulae, by Theorem 55. Here we address the general question by using classical model-theoretic tools and the constructions introduced earlier in this section.

Since modal formulae express first-order conditions on (pointed) Kripke structures, these are special cases of *first-order definable* (by a single first-order sentence), respectively *elementary* (definable by any set of first-order sentences) classes and properties. Keisler's theorem [12, Theorem 4.1.12] characterising elementary and first-order definable classes is therefore relevant here: a class of first-order structures is elementary iff it is closed under elementary equivalence and ultraproducts; it is first-order definable iff both the class and its complement are elementary. Since modal formulae cover only a fragment of the first-order language $\mathrm{FO}(\tau_\Phi)$, these results give necessary but not sufficient conditions for modal definability of classes of (pointed) Kripke structures. But 'elementary equivalence' for modal logic is modal equivalence. Would that adjustment of Keisler's theorem suffice to guarantee modal definability? The answer is 'yes' in both cases. The following is from [22].

THEOREM 94. *A class $\mathcal{K}$ of (pointed) Kripke structures is definable by a set of modal formulae iff it is closed under modal equivalence and ultraproducts; $\mathcal{K}$ is definable by a single modal formula iff both $\mathcal{K}$ and its complement are definable by a set of modal formulae.*

Proof. These can be proved by adapting the proof of Keisler's theorem. Alternatively, we may invoke a corollary of the Keisler–Shelah theorem (cf. Corollary 6.1.16 and Theorem 6.1.15 in [12]) which states that a class of first-order structures is elementary iff it is closed under isomorphism and ultraproducts while its complement is closed under ultrapowers. The latter condition here follows from closure under modal equivalence. Once $\mathcal{K}$ has been shown to be elementary, a general argument can be applied that works not only for modal formulae but for any other natural fragment Δ of first-order logic (see [12, Lemma 3.2.1]): if $\Delta \subseteq \mathrm{FO}$ is closed under negation and disjunction, then an elementary class is axiomatisable with formulae from Δ iff it is closed under Δ-equivalence.

For definability by a single formula, one may use compactness for ML just as for FO to show that whenever both the given class and it complement are definable by a set of formulae, then the class (and its complement) are definable by a single formula. Alternatively, one may first establish first-order definability of $\mathcal{K}$, and then use Theorem 55 and

bisimulation invariance to see that the defining formula must be equivalent to a modal formula. ❑

Note that, as an immediate consequence of (the classical proof of) Theorem 55, an elementary class of (pointed) Kripke structures is closed under modal equivalence iff it is closed under bisimulations. Therefore, we can strengthen somewhat the results above, by replacing closure under modal equivalence by bisimulation closure, but at the expense of demanding closure of the complement under ultrapowers. See [22] and [5, Theorems 2.75, 2.76] for the following.

THEOREM 95. *For any class $\mathcal{K}$ of pointed Kripke structures:*

(i) *$\mathcal{K}$ is definable by a set of modal formulae iff it is closed under bisimulation and ultraproducts, while its complement is closed under ultrapowers.*

(ii) *$\mathcal{K}$ is definable by a single modal formula iff it is closed under bisimulation, while both it and its complement are closed under ultraproducts.*

Proof. For the non-trivial part of (i): assuming the closure conditions for $\mathcal{K}$ and its complement, we consider the modal theory $\mathrm{Th}_{\mathrm{ML}}(\mathcal{K})$ and show that it defines $\mathcal{K}$, i.e., every model of it is in $\mathcal{K}$. For details see [22], [5, Theorem 2.75]. Alternatively, we can take a shortcut: by Theorem 89 the closure conditions imply that $\mathcal{K}$ is closed under modal equivalence, and hence Theorem 94 applies.

For the non-trivial part of (ii) we may use (i) and a standard compactness argument as in the proof of Keisler's theorem (see [22] and [5, Theorems 2.76]), or use Theorem 94 again. ❑

Similar results can be obtained for classes of Kripke structures; we leave these to the reader.

Finally, we mention the following results of Venema [138] which characterise modal definability of classes of (pointed) Kripke structures in purely modal terms, i.e., without involving the typical constructions from classical logic. In what follows, a bisimulation $\rho\colon \mathfrak{M} \rightleftarrows \mathfrak{M}'$ is *surjective* if every state in $\mathfrak{M}'$ has a bisimilar one in $\mathfrak{M}$; an *ultrafilter union* of a family of pointed Kripke structures $\left\{\mathfrak{M}^i, w^i\right\}_{i\in I}$ is a pointed Kripke structure $\left(\mathfrak{ue}\left(\biguplus_{i\in I}\mathfrak{M}^i\right), \mathsf{w}\right)$, where w is an ultrafilter containing every co-finite subset of $\{w_i \mid i \in I\}$.

THEOREM 96. *A class of Kripke structures is modally definable iff it is closed under disjoint unions, surjective bisimulations, and ultrafilter extensions, while it reflects ultrafilter extensions.*

A class of pointed Kripke structures is modally definable iff it is closed under bisimulations and ultrafilter unions, and reflects ultrafilter extensions.

To summarise: model theory of modal logic over Kripke structures essentially derives from first-order model theory, with the crucial extra feature of bisimulation invariance. The additional requirement of bisimulation invariance leads us from classical model theory to modal model theory and allows us to develop the analogy between them further.

7 GENERAL FRAMES

Neither of the two kinds of semantic structures we have considered so far, viz. Kripke frames and Kripke structures, provides a completely satisfactory framework for the se-

mantics for modal logic. On the one hand, truth and validity in Kripke structures, with its crucial dependency on given valuations, does not reflect the richer semantics in terms of validity in frames. On the other hand, validity in frames, being an essentially second-order notion, is in general deductively intractable. As a consequence, frame-incomplete modal logics are the rule, rather than the exception (see Chapter 7 of this handbook). It is therefore necessary to look for a new type of semantic structures, 'hybrids' between Kripke structures and frames, combining the expressive richness of the frame-based semantics with the flexibility and good deductive behaviour of the one based on Kripke structures.

Such structures, called *general frames*, were introduced in modal logic by Thomason in [124], with precursors in [97] and [28]. General frames are analogues to Henkin's 'general models' for second-order logic, extending first-order structures with a family of 'admissible sets', and restricting the second-order quantification to such sets only. Independently, general frames essentially arose from the seminal study by Jónsson and Tarski [78] of Boolean algebras with operators (see also Chapter 6 of this handbook), since they appear as the 'concrete', set-theoretic counterparts of modal algebras, arising in the Jónsson–Tarski representation theorem, and thus providing the link between the algebraic and relational semantics.

In this section we introduce the modal semantics based on general frames, develop the basic model theory of general frames and briefly mention the *duality theory* which relates them to algebras. We then discuss the relevance and use of general frames to the model theory of the frame-based modal semantics, in terms of *persistence* of modal formulae with respect to various important classes of general frames.

7.1 General frames as semantic structures in modal logic

Note that the operators $\langle R\rangle$ and $[R]$ defined in section 1.2 are monotone. Besides, the operators $\langle R\rangle$ are *normal* (preserving falsum) and *additive* (distributive over disjunctions); see Chapter 6 of this handbook. Hence every structure $\langle \mathcal{P}(W); \cap, -, \varnothing, \{\langle R_\alpha\rangle\}_{\alpha\in\tau}\rangle$ is a (complete and atomic) Boolean algebra with operators in the terms of [78] (see also Chapter 6), called a *modal τ-algebra.*

DEFINITION 97. Given a τ-frame $\mathfrak{F} = \langle W, \{R_\alpha\}_{\alpha\in\tau}\rangle$, a *general τ-frame over* $\mathfrak{F}$ is a structure $\langle \mathfrak{F}, \mathbb{W}\rangle$ expanding $\mathfrak{F}$ with a τ-algebra of *admissible subsets* of $\mathcal{P}(W)$, closed under boolean operations and the operators $\{\langle R_\alpha\rangle\}_{\alpha\in\tau}$, i.e., $\mathbb{W}$ is a τ-subalgebra of $\langle \mathcal{P}(W); \cap, -, \varnothing, \{\langle R_\alpha\rangle\}_{\alpha\in\tau}\rangle$.

Given a general τ-frame $\mathfrak{G} = \langle \mathfrak{F}, \mathbb{W}\rangle$ we denote $\mathfrak{F}$ by $\mathfrak{G}_\#$ and the τ-algebra $\mathbb{W}$ by $\mathfrak{G}^+$.

EXAMPLE 98. For every Kripke structure $\mathfrak{M} = \langle \mathfrak{F}, V\rangle$, $\langle \mathfrak{F}, \{V(\varphi) \mid \varphi \in \mathrm{ML}(\tau)\}\rangle$ is a general τ-frame over $\mathfrak{F}$, *generated by* $\mathfrak{M}$. In particular, the general τ-frame $\mathfrak{G}_L$ generated by the canonical Kripke structure $\mathfrak{M}_L$ (see Chapter 7 of this handbook) of a normal modal logic L is called the *canonical general frame of* L.

Among the general frames over $\mathfrak{F} = \langle W, \{R_\alpha\}_{\alpha\in\tau}\rangle$ there is a least one, viz. $\mathfrak{F}_{\min} = \langle \mathfrak{F}, \mathbb{W}_{\min}\rangle$ generated from the Kripke structure $\mathfrak{M}_{\min} = \langle \mathfrak{F}, V_{\min}\rangle$ where $V_{\min}(p) = \emptyset$ for every $p \in \Phi$, and a greatest one, viz. the *full general τ-frame* $\mathfrak{F}_{\max} = \langle \mathfrak{F}, \mathcal{P}(W)\rangle$. Clearly, local (as well as global) validity in $\mathfrak{F}$ and $\mathfrak{F}_{\max}$ coincide. So we can safely identify the τ-frame $\mathfrak{F}$ with $\mathfrak{F}_{\max}$. Furthermore, the family of all general frames over a τ-frame $\mathfrak{F} = \langle W, \{R_\alpha\}_{\alpha\in\tau}\rangle$ forms a complete lattice.

DEFINITION 99. Given a general τ-frame $\mathfrak{G} = \langle \mathfrak{F}, \mathbb{W} \rangle$, a *valuation over* $\mathfrak{G}$ is any valuation $V : \Phi \to \mathbb{W}$. A Kripke structure $\langle \mathfrak{F}, V \rangle$ where V is a valuation over $\mathfrak{G}$ is a *Kripke structure over* $\mathfrak{G}$, also denoted by $\langle \mathfrak{G}, V \rangle$ or $\langle \mathfrak{F}, \mathbb{W}, V \rangle$.

It follows by a routine induction that if $\mathfrak{M} = \langle \mathfrak{F}, \mathbb{W}, V \rangle$, then $V(\varphi) \in \mathbb{W}$ for every $\varphi \in \mathrm{ML}(\tau)$.

DEFINITION 100. Given a formula $\varphi \in \mathrm{ML}(\tau)$, a general τ-frame $\mathfrak{G}$, and $w \in W$, we say that *φ is (locally) valid at w in* $\mathfrak{G}$, denoted $\mathfrak{G}, w \models \varphi$, if φ is true at w in every Kripke structure over $\mathfrak{G}$. *φ is valid in* $\mathfrak{G}$, denoted $\mathfrak{G} \models \varphi$, if φ is valid in $\mathfrak{G}$ at every $w \in W$, i.e., φ is valid in every Kripke structure over $\mathfrak{G}$.

Note that local validity of modal formulae in a general τ-frame is preserved under the rule Modus Ponens and under taking uniform substitutions, while validity is also preserved under Necessitation.

All general frames generated from Kripke structures have an at most countable algebra of admissible sets, so not every general frame is of that type. On the other hand, every general frame *can* be generated from a Kripke structure in an extended language with an appropriately large cardinality of the set of atomic propositions. This observation is sufficient to transfer various results and constructions from Kripke structures to general frames.

However, as semantic structures for modal logic, general frames match most closely modal algebras. Indeed, as already noted, every general τ-frame $\mathfrak{G}$ generates a '*complex τ-algebra*' $\mathfrak{G}^+$. Conversely, every τ-algebra $\mathfrak{A}$ determines a general frame $\mathfrak{A}_+$ based on the *ultrafilter frame* of that algebra (see section 7.2), and is moreover embedded in $(\mathfrak{A}_+)^+$ in a way extending the Stone representation for Boolean algebras. That embedding is the subject of the celebrated *Jónsson–Tarski representation theorem* (see [78], [5, Section 5.3], or Chapter 6 of this handbook). Furthermore, there exists an algebraic-categorial *duality* between general frames and modal algebras, systematically developed by Goldblatt in [43, 44, 47] and later, from a topological perspective by [114] (see also [5, Section 5.4]), discussed in detail in Chapter 6 of this handbook.

7.2 Constructions and truth preservation results on general frames

Bisimulations and special cases

DEFINITION 101. Let $\mathfrak{G} = \langle \mathfrak{F}, \mathbb{W} \rangle$ and $\mathfrak{G}' = \langle \mathfrak{F}', \mathbb{W}' \rangle$ be two general τ-frames. A bisimulation ρ between $\mathfrak{F}$ and $\mathfrak{F}'$ is a *bisimulation between* $\mathfrak{G}$ *and* $\mathfrak{G}'$ if for every valuation V over $\mathfrak{G}$ there is a valuation V' over $\mathfrak{G}'$ such that $\rho\colon \langle \mathfrak{G}, V \rangle \rightleftarrows \langle \mathfrak{G}', V' \rangle$, and vice versa.

A bisimulation between pointed general frames is defined likewise.

Note that not every bisimulation between Kripke frames is a bisimulation between them as full general frames, because not every valuation over one of them must have a matching valuation satisfying *atom equivalence.*

COROLLARY 102. *If $\rho\colon (\mathfrak{G}, w) \rightleftarrows (\mathfrak{G}', w')$ is a bisimulation between pointed general τ-frames $(\mathfrak{G}, w)$ and $(\mathfrak{G}', w')$ then $(\mathfrak{G}, w) \equiv_{\mathrm{ML}} (\mathfrak{G}', w')$. Likewise, if $\rho\colon \mathfrak{G} \rightleftarrows \mathfrak{G}'$, then $\mathfrak{G} \equiv_{\mathrm{ML}} \mathfrak{G}'$.*

The definitions of generated subframes, bounded morphisms, and disjoint unions can be extended to general frames.

DEFINITION 103. Given a general τ-frame $\mathfrak{G} = \langle \mathfrak{F}, \mathbb{W} \rangle$, a *generated subframe of* $\mathfrak{G}$ is any general τ-frame $\mathfrak{G}' = \langle \mathfrak{F}', \mathbb{W}' \rangle$ where $\mathfrak{F}' \trianglelefteq \mathfrak{F}$ and $\mathbb{W}' = \{X \cap \mathrm{dom}(\mathfrak{F}') \mid X \in \mathbb{W}\}$.

DEFINITION 104. Let $\mathfrak{G} = \langle \mathfrak{F}, \mathbb{W} \rangle$ and $\mathfrak{G}' = \langle \mathfrak{F}', \mathbb{W}' \rangle$ be two general τ-frames and $\rho \colon \mathfrak{F} \xrightarrow{\rightleftarrows} \mathfrak{F}'$ a bounded morphism. Then ρ is a *bounded morphism* from $\mathfrak{G}$ to $\mathfrak{G}'$ if for every $Y \in \mathbb{W}'$, $\rho^{-1}[Y] \in \mathbb{W}$; ρ is a *bounded strong morphism* from $\mathfrak{G}$ to $\mathfrak{G}'$ if it is a bounded morphism from $\mathfrak{G}$ to $\mathfrak{G}'$ and for every $X \in \mathbb{W}$, $\rho[X] \in \mathbb{W}'$ and $X = \rho^{-1}[\rho[X]]$.

DEFINITION 105. The *disjoint union* of the family $\left\{\mathfrak{G}^i = \left\langle \mathfrak{F}^i, \mathbb{W}^i \right\rangle\right\}_{i \in I}$ of general τ-frames is $\biguplus^i_{i \in I} \mathfrak{G}^i = \left\langle \biguplus_{i \in I} \mathfrak{F}^i, \mathbb{W} \right\rangle$, where $\mathbb{W} = \left\{\biguplus_{i \in I} X^i \mid X^i \in \mathbb{W}^i \text{ for each } i \in I\right\}$.

We leave it to the reader to check that generated subframes and disjoint unions of general frames produce general frames indeed, and to see that they, as well as bounded strong morphisms, are particular cases of general frame bisimulations. The associated preservation results are immediate, and are left to the reader, too. As for bounded morphisms of general frames, in general they are not general frame bisimulations and only preserve validity in the forward direction.

Ultrafilter extensions and ultraproducts

The construction of ultrafilter extensions of frames can be generalised to the Stone representation of modal algebras (see Chapter 6 of this handbook), which in turn are essentially general frames, thus defining ultrafilter extensions of general frames. More precisely, given a general τ-frame $\mathfrak{G} = \langle W, \{R_\alpha\}_{\alpha \in \tau}, \mathbb{W} \rangle$ over a frame $\mathfrak{F}$, let $\mathbf{U}(\mathbb{W})$ be the set of all ultrafilters over the algebra $\mathfrak{G}^+$. For each $\alpha \in \tau$ we define a binary relation $R^{\mathbb{W}}_\alpha$ on $\mathbf{U}(\mathbb{W})$ just like $R^{\mathfrak{ue}}_\alpha$ in $\mathfrak{ue}(\mathfrak{F})$, i.e., for any $\mathsf{u}, \mathsf{w} \in \mathbf{U}(\mathbb{W})$:

$$\mathsf{u} R^{\mathbb{W}}_\alpha \mathsf{w} \text{ iff } \langle R_\alpha \rangle (X) \in \mathsf{u} \text{ for every } X \in \mathsf{w}.$$

The frame $(\mathfrak{G}^+)_+ = \left\langle \mathbf{U}(\mathbb{W}), \{R^{\mathbb{W}}_\alpha\}_{\alpha \in \tau} \right\rangle$ is called the *ultrafilter frame of the τ-algebra* $\mathfrak{G}^+$.

Finally, we put $\mathbb{W}^{\mathfrak{ue}} := \{\mathfrak{u}_{\mathbb{W}}(X) \mid X \in \mathbb{W}\}$ where $\mathfrak{u}_{\mathbb{W}}(X) = \{\mathsf{u} \in \mathbf{U}(\mathbb{W}) \mid X \in \mathsf{u}\}$. It is routine to check that $\left\langle R^{\mathbb{W}}_\alpha \right\rangle (\mathfrak{u}_{\mathbb{W}}(X)) = \mathfrak{u}_{\mathbb{W}}(\langle R_\alpha \rangle (X))$ and hence $\mathbb{W}^{\mathfrak{ue}}$ is a modal τ-algebra.

DEFINITION 106. Given a general τ-frame $\mathfrak{G} = \langle W, \{R_\alpha\}_{\alpha \in \tau}, \mathbb{W} \rangle$, *the ultrafilter extension of* $\mathfrak{G}$ is the general τ-frame $\mathfrak{ue}(\mathfrak{G}) := \left\langle \mathbf{U}(\mathbb{W}), \{R^{\mathbb{W}}_\alpha\}_{\alpha \in \tau}, \mathbb{W}^{\mathfrak{ue}} \right\rangle$, also known as the *general ultrafilter frame* of *the τ-algebra* $\mathfrak{G}^+$.

From the basic properties of ultrafilters, and the closure of $\mathbb{W}^{\mathfrak{ue}}$ under $\left\langle R^{\mathbb{W}}_\alpha \right\rangle$, it follows that $\mathfrak{G}^+ \cong \mathfrak{ue}(\mathfrak{G})^+$ for any general τ-frame $\mathfrak{G}$. Note, however, that $\mathfrak{ue}(\mathfrak{G}) \cong \mathfrak{G}$ does not hold in general, and in section 7.3 we will characterise the general frames for which this is the case. Still, since validity of modal formulae in $\mathfrak{G}$ and in $\mathfrak{G}^+$ coincide, we obtain the following.

THEOREM 107. *For any general τ-frame $\mathfrak{G}$, $\mathfrak{ue}(\mathfrak{G}) \equiv_{\mathrm{ML}} \mathfrak{G}$.*

DEFINITION 108. Let $\{\mathfrak{G}^i = \langle \mathfrak{F}^i, \mathbb{W}^i \rangle\}_{i \in I}$ be a family of general τ-frames indexed by a set I. For any ultrafilter $\mathbf{U}$ on I, the *ultraproduct of* $\{\mathfrak{G}^i\}_{i \in I}$ *over* $\mathbf{U}$ is the general τ-frame $\prod^{\mathbf{U}}_{i \in I} \mathfrak{G}^i := \langle \prod^{\mathbf{U}}_{i \in I} \mathfrak{F}^i, \mathbb{W}^{\mathbf{U}} \rangle$, where $\mathbb{W}^{\mathbf{U}} = \{\prod^{\mathbf{U}}_{i \in I} X^i \mid X^i \in \mathbb{W}^i \text{ for each } i \in I\}$.

Note that the ultraproduct of a family of Kripke frames regarded as full general frames is *not* a full general frame itself, so it differs from the ultraproduct of frames, as defined

earlier. To distinguish these, we call the former *general ultraproduct* of frames. Unlike the latter, every valuation in it is an ultraproduct of respective valuations in the components, whence the following preservation result (see [43, 44, 47]).

PROPOSITION 109. *For every family of general τ-frames $\left\{\mathfrak{G}^i = \left\langle \mathfrak{F}^i, \mathbb{W}^i \right\rangle\right\}_{i\in I}$, ultrafilter $\mathbf{U}$ on I, element $\mathbf{w_U} \in \prod_{i\in I}^{\mathbf{U}} \mathfrak{F}^i$, and formula φ of ML(τ) :*

(i) $\prod_{i\in I}^{\mathbf{U}} \mathfrak{G}^i, \mathbf{w_U} \models \varphi$ *iff* $\{j \in I \mid \mathfrak{G}^j, \mathbf{w}(j) \models \varphi\} \in \mathbf{U}$.

(ii) $\prod_{i\in I}^{\mathbf{U}} \mathfrak{G}^i \models \varphi$ *iff* $\{j \in I \mid \mathfrak{G}^j \models \varphi\} \in \mathbf{U}$.

7.3 Special types of general frames and persistence of modal formulae

Let $\mathcal{G}$ be the class of all general τ-frames of a fixed modal type τ, and let $\mathcal{C}$ be any subclass of $\mathcal{G}$.

DEFINITION 110. A formula $\varphi \in \mathrm{ML}(\tau)$ is *locally $\mathcal{C}$-persistent*, if for every general τ-frame $\mathfrak{G} = \langle \mathfrak{F}, \mathbb{W} \rangle \in \mathcal{C}$, and $w \in \mathrm{dom}(\mathfrak{F})$, $\mathfrak{G}, w \models \varphi$ implies $\mathfrak{F}, w \models \varphi$; φ is *$\mathcal{C}$-persistent*, if for every general τ-frame $\mathfrak{G}= \langle \mathfrak{F}, \mathbb{W} \rangle \in \mathcal{C}$, $\mathfrak{G} \models \varphi$ implies $\mathfrak{F} \models \varphi$.

Clearly, local persistence implies persistence, but the converse does not always hold. While often the practically important notion is the latter, the former is more natural.

A general frame can be thought of as a frame in which a restriction on the valuations is imposed by allowing only those valuations which assign admissible sets to the propositional variables (and hence, to all formulae). Thus, the idea of persistence is that it enables one to conclude (local) validity, i.e., truth under *every* valuation, of a modal formula in a frame, based on its truth under *some* special valuations, viz. the admissible ones. In other words, a formula is $\mathcal{C}$-persistent if, whenever it is falsified in a Kripke frame $\mathfrak{F}$, it is falsified by *some* admissible valuation in *each* general frame from $\mathcal{C}$ over $\mathfrak{F}$. Thus, persistence gives a measure of the 'semantic complexity' of a formula, in terms of its falsifying valuations. Note that a modal formula is locally $\mathcal{G}$-persistent iff it is semantically equivalent to a constant formula (i.e., a formula without propositional variables). Indeed, every constant formula is $\mathcal{G}$-persistent. Conversely, if φ is $\mathcal{G}$-persistent, then for every pointed frame $(\mathfrak{F}, w)$, $\mathfrak{F}, w \models \varphi$ iff $\langle \mathfrak{F}, V_\perp \rangle, w \models \varphi$, where $V_\perp$ assigns $\emptyset$ to every atomic proposition, iff $\mathfrak{F}, w \models \varphi_\perp$ where $\varphi_\perp$ is obtained from φ by replacing all atomic propositions by $\perp$.

We will introduce some important classes of general frames, persistence with respect to which provides sufficient conditions for good expressive or axiomatic behaviour of the formulae.

DEFINITION 111. Let $\mathfrak{G}= \langle W, \{R_\alpha\}_{\alpha\in\tau}, \mathbb{W} \rangle$ be a general τ-frame and $\alpha \in \tau$. The relation R_α is *tight* in $\mathfrak{G}$ if for every $u, w \in W$: $uR_\alpha w$ iff for all $X \in \mathbb{W}$, $w \in X$ implies $u \in \langle R_\alpha \rangle(X)$; equivalently, iff $u \in \bigcap\{\langle R_\alpha \rangle(X) \mid X \in \mathbb{W} \text{ and } w \in X\}$.

Recall, for the compactness property below, that a family of sets $\mathcal{F}$ has the *finite intersection property* (FIP) if the intersection of every finite sub-family of $\mathcal{F}$ is non-empty.

DEFINITION 112. A general τ-frame $\langle W, \{R_\alpha\}_{\alpha\in\tau}, \mathbb{W} \rangle$ is:

- *differentiated*, if for every $u, u' \in W$, if $u \neq u'$ then there is $X \in \mathbb{W}$ such that $u \in X$ and $u' \notin X$;
- *tight*, if R_α is tight for every $\alpha \in \tau$;

– *discrete*, if $\{u\} \in \mathbb{W}$ for every $u \in W$;
– *elementary*, if every subset of W that is $\mathrm{FO}(\tau)$-definable with parameters (in the sense of Definition 39) is admissible;
– *compact*, if every family of admissible sets in $\mathfrak{G}$ with FIP has a non-empty intersection;[12]
– *refined*, if it is differentiated and tight;
– *descriptive*, if it is refined and compact.

Amongst all discrete general frames over a Kripke frame $\mathfrak{F}$, there is a *least* one, viz. $\mathfrak{D}(\mathfrak{F})$, generated from all singletons by closing under the Boolean and modal operators. It contains all finite and co-finite sets in $\mathfrak{F}$. Likewise, amongst all elementary general frames over a Kripke frame $\mathfrak{F}$, there is a *least* one, viz. $\mathfrak{E}(\mathfrak{F})$, in which the admissible sets are precisely the subsets of the domain of $\mathfrak{F}$ that are parametrically first-order definable in $\mathrm{FO}(\tau)$.

Assuming the type τ is fixed, the class of all differentiated (resp. tight, discrete, elementary, refined, descriptive) general τ-frames will be denoted by $\mathcal{DF}$ (resp. $\mathcal{T}, \mathcal{DI}, \mathcal{E}, \mathcal{R}, \mathcal{D}$).

Here are some relationships between these classes.

- Every full general frame is discrete, and therefore, refined (see below). Every finite, but no infinite, discrete general frame is descriptive, for otherwise the intersection of all sets $W \setminus \{w\}$ would have to be non-empty; on the other hand, every finite differentiated frame is full.

- Every discrete frame is refined. Indeed, for tightness note that in every discrete frame $xR_\alpha w$ holds iff $x \in \langle R_\alpha \rangle(\{w\})$. The converse need not hold, e.g., canonical general frames (see Chapter 7 of this handbook) are refined, even descriptive, but not discrete, being infinite.

- Every elementary frame is discrete, while the converse does not hold, as we will see further.

To summarise: $\mathcal{E} \subsetneqq \mathcal{DI} \subsetneqq \mathcal{R} = \mathcal{DF} \cap \mathcal{T}$; $\mathcal{D} \subsetneqq \mathcal{R}$; $\mathcal{D} \nsubseteq \mathcal{DI} \nsubseteq \mathcal{D}$.

Below, we list some remarks on the various notions of persistence and relationships between them. Analogous remarks apply to local persistence.

- First, note that if $\mathcal{C}_1 \subseteq \mathcal{C}_2$, then $\mathcal{C}_2$-persistence implies $\mathcal{C}_1$-persistence.

- A formula is $\mathcal{DI}$-persistent iff it is valid in a frame $\mathfrak{F}$ whenever it is valid in $\mathfrak{D}(\mathfrak{F})$. Likewise, a formula is $\mathcal{E}$-persistent iff it is valid in a frame $\mathfrak{F}$ whenever it is valid in $\mathfrak{E}(\mathfrak{F})$.

- While every (locally) $\mathcal{R}$-persistent formula is $\mathcal{DI}$-persistent, the converse does not hold, a simple witness being e.g., the 'density' formula $\Diamond p \to \Diamond\Diamond p$ (see [5, p.319]).

- Also, not every (even locally) $\mathcal{D}$-persistent formula is $\mathcal{DI}$-persistent (and hence, even less $\mathcal{R}$-persistent), a witness being *Geach's formula* $\Diamond\Box p \to \Box\Diamond p$, defining the Church–Rosser confluence property of the accessibility relation (see [5, p.305]).

[12] This is equivalent to the requirement that every ultrafilter over $\mathfrak{G}^+$ consists of all admissible sets containing a fixed state in $\mathfrak{G}$.

Moreover, not every $\mathcal{D}$-persistent formula is $\mathcal{E}$-persistent, as we will see in section 8.2.

- Not every (even locally) $\mathcal{E}$-persistent is $\mathcal{DI}$ -persistent, again witnessed by Geach's formula.

- Finally, not every (even locally) $\mathcal{DI}$-persistent formula is $\mathcal{D}$-persistent. The formula $\mathbf{vB} = \Box\Diamond\top \to \Box(\Box(\Box p \to p) \to p)$, proposed by van Benthem in [127], is an example. First, note that for every discrete general frame $\mathfrak{G} = \langle \mathfrak{F}, \mathbb{W}\rangle$ and $w \in \mathrm{dom}(\mathfrak{F})$, $\mathfrak{G}, w \models \mathbf{vB}$ implies $\mathfrak{G}, w \models \Box\Diamond\top \to \Box\bot$; hence $\mathfrak{G}_{\#}, w \models \Box\Diamond\top \to \Box\bot$. Indeed, assuming $\mathfrak{G}, w \models \Box\Diamond\top \wedge \neg\Box\bot$, for any successor u of w the valuation $W\backslash\{u\}$ for p falsifies $\mathbf{vB}$ at w. Furthermore, for every frame $\mathfrak{F}$ and $w \in \mathrm{dom}(\mathfrak{F})$, $\mathfrak{F}, w \models \Box\Diamond\top \to \Box\bot$ implies $\mathfrak{F}, w \models \mathbf{vB}$. Hence $\mathbf{vB}$ is locally $\mathcal{DI}$-persistent. On the other hand, $\mathbf{vB}$ is not $\mathcal{D}$-persistent. Indeed, as shown in [127] (see also [5, p.216]) $\mathbf{vB}$ is valid in a certain general frame $\mathfrak{J}$, the modal logic $\mathbf{KvB}$ of which is incomplete. That is because $\Box\Diamond\top \to \Box\bot$, not being valid in $\mathfrak{J}$, is not a theorem of $\mathbf{KvB}$ while, as seen above, it is valid in every frame for $\mathbf{KvB}$. Thus, while $\mathbf{vB}$ is valid in the (descriptive) canonical frame of $\mathbf{KvB}$, it fails in the underlying Kripke frame which falsifies $\Box\Diamond\top \to \Box\bot$.

  Consequently, not every locally $\mathcal{E}$-persistent formula is $\mathcal{D}$-persistent.

To summarise again, if we denote by $\mathcal{C}^p$ the set of all $\mathcal{C}$-persistent formulae, we have the following: $\mathcal{DF}^p \cap \mathcal{T}^p = \mathcal{R}^p \subsetneq \mathcal{DI}^p \subsetneq \mathcal{E}^p$; $\mathcal{R}^p \subsetneq \mathcal{D}^p$; $\mathcal{DI}^p \nsubseteq \mathcal{D}^p \nsubseteq \mathcal{E}^p$.

The same relationships hold for local persistence.

Now, we discuss some important results about refined and descriptive frames and the related persistence properties, while elementary frames and elementary persistence will be discussed in section 8.2.

First, note ([124]) that every general frame $\mathfrak{G} = \langle \mathfrak{F}, \mathbb{W}\rangle$ can be 'refined' by constructing a refined quotient of it over the set $W^{\sim}$ of all equivalence classes modulo the equivalence relation $\sim$, defined as $v \sim w$ iff $\forall X \in \mathbb{W}(v \in X \iff w \in X)$, and taking as admissible all sets of the type $X^{\sim} = \{w^{\sim} \mid w \in X\}$ for $X \in \mathbb{W}$. It now remains to 'tighten' all accessibility relations by closing under the definition of tightness: for every $u^{\sim}, w^{\sim} \in W^{\sim}$, $u^{\sim} R_{\alpha}^{\sim} w^{\sim}$ holds iff for all $X^{\sim} \in \mathbb{W}^{\sim}$ and $u' \sim u, w' \sim w$, if $w' \in X$ then $u' \in \langle R_{\alpha}\rangle(X)$. Note, however, that (see [10, p.263]) while for finite frames this construction produces a bounded morphic image, this is not necessarily the case when applied to infinite general frames.

Descriptive frames typically appear as the *canonical general frames* (see Chapter 7 of this handbook) of every normal modal logic without any special inference rules. Thus, all $\mathcal{D}$-persistent formulae are valid in the underlying canonical Kripke frames, and hence they axiomatise Kripke complete logics. For that reason the $\mathcal{D}$-persistent formulae are also called *canonical*.[13] However, in hybrid logics with nominals (see Chapter 14 of this handbook) or in logics with special additional rules of inference, e.g., the non-ξ rules in [137], $\mathcal{D}$-persistent formulae *need not* be canonical, because the canonical general frames

[13] Note that across the literature on modal logic the term 'canonicity' is used in somewhat different, and not entirely equivalent, senses (see [126, 127]). For instance, Fine defines in [28] canonicity of a set of formulae as validity of every formula of that set in any canonical frame built for a modal language with any cardinality of propositional variables. Since all canonical models generate descriptive frames, the notion of canonicity adopted here following [126] is at least as strong as Fine's.

for such logics are only discrete (for hybrid logics) or refined (in logics with additional 'context' rules, see [52]). In such cases, $\mathcal{DI}$-persistence or $\mathcal{R}$-persistence is the right notion of canonicity. $\mathcal{DI}$-persistent formulae have the important property to remain canonical when added as axioms to hybrid logics with nominals, while $\mathcal{R}$-persistent formulae remain canonical not only in the presence of other axioms, but even if additional rules of inference of the type mentioned above are added to the axiomatic system.

Descriptive frames feature prominently in the duality theory between general frames and modal algebras, as they turn out to be precisely the fixed points of ultrafilter extensions of general frames, which are essentially the Stone representations of modal algebras (see Chapter 6 of this handbook).

PROPOSITION 113. *A general τ-frame $\mathfrak{G}$ is descriptive iff $\mathfrak{G} \cong \mathfrak{ue}(\mathfrak{G})$.*

Indeed, the proof that every ultrafilter extension is descriptive is just a variation of the proof that every canonical general frame is descriptive (see Chapter 7 of this handbook). For the converse, the crucial observation is that, given a descriptive general frame $\mathfrak{G} = \langle \mathfrak{F}, \mathbb{W} \rangle$, for every $w \in \mathbf{F}$, the set $\mathsf{u}_{\mathbb{W}}[w] = \{X \in \mathbb{W} \mid w \in X\}$ is an ultrafilter in $\mathbb{W}$, and every ultrafilter in $\mathbb{W}$, due to the compactness of $\mathfrak{G}$, is of this type. Thus, the mapping $\lambda w.\mathsf{u}_{\mathbb{W}}[w]$ is a bijection (since $\mathfrak{G}$ is differentiated) between $\mathfrak{G}$ and $\mathfrak{ue}(\mathfrak{G})$. This bijection is in fact an isomorphism, due to the tightness of $\mathfrak{G}$.

Consequently, by Theorem 107, every general frame is modally equivalent to a descriptive frame. Therefore, every $\mathcal{D}$-persistent formula φ preserves its validity from a frame $\mathfrak{F}$ to the ultrafilter extension of the full general frame $\mathfrak{F}_{\max}$, which is based on $\mathfrak{ue}(\mathfrak{F})$. Since $\mathfrak{ue}(\mathfrak{F}_{\max})$ is descriptive, by $\mathcal{D}$-persistence, φ preserves validity from $\mathfrak{F}$ to $\mathfrak{ue}(\mathfrak{F})$. Conversely, if φ preserves validity in ultrafilter extensions of frames, then it is $\mathcal{D}$-persistent by Theorem 115. Thus, we obtain:

PROPOSITION 114. *A modal formula is (locally) $\mathcal{D}$-persistent iff its validity is (locally) preserved in ultrafilter extensions of frames.*

Every general τ-frame $\mathfrak{G} = \langle W, \{R_\alpha\}_{\alpha\in\tau}, \mathbb{W} \rangle$ determines a *topological space* $T(\mathfrak{G})$ with a base of clopen sets $\mathbb{W}$, and a set of closed sets denoted by $\mathbf{C}(\mathbb{W})$. For a detailed study of this topology, its properties and applications in modal logic see [114]. Hereafter, a *closed* set in the general τ-frame $\mathfrak{G}$ will mean a subset of the domain closed with respect to the topology $T(\mathfrak{G})$, i.e., an intersection of a family of admissible sets.

A number of important properties of general frames can be phrased in terms of their topology. For instance, in every discrete frame $\mathfrak{G}$ the topology $T(\mathfrak{G})$ is discrete. Indeed, every non-empty set is a union of its singleton subsets, which are open in $T(\mathfrak{G})$; hence every subset of $\mathfrak{G}$ is open. Also, differentiatedness of a general frame is equivalent to T_2-separability (Hausdorffness) of its topology, while compactness, as defined above, is equivalent to the standard topological notion of compactness. Thus, for any compact and differentiated τ-frame $\mathfrak{G}$, $T(\mathfrak{G})$ is a compact Hausdorff space.

Finally, it is instructive to explore which constructions on general frames preserve each of the classes discussed above. For instance, differentiatedness, tightness, and discreteness are preserved in generated subframes and disjoint unions, while compactness is not. Conversely, bounded morphisms preserve compactness, but not discreteness, differentiatedness and tightness. Besides, discreteness, differentiatedness, and tightness (and hence, refinedness), being properties definable in a suitable first-order language for states and admissible sets, and membership between them, are preserved in ultraproducts, while

descriptiveness is preserved in finite disjoint unions, but never in infinite ones, nor necessarily in ultraproducts [44, 47].

How does persistence determine the expressiveness of a formula? We will discuss this issue in section 8.2 in connection with first-order definability of modal formulae.

Before closing this section, let us highlight again the role of general frames in the modal theory of modal logic:

- general frames provide a natural link between the first-order semantics on Kripke structures and the second-order semantics on frames, and are thus analogous to Henkin's general models for second-order logic.
- general frames are essentially equivalent to modal algebras, via the duality theory outlined in Chapter 6 of this handbook, and thus provide algebraic semantics for modal logic.
- the notion of persistence of (the truth/validity of) modal formulae with respect to natural classes of general frames is instrumental in characterising their model-theoretic behaviour.

8 MODAL LOGIC ON FRAMES

So far we have mainly studied modal logic as a fragment of first-order logic over Kripke structures. In this section we discuss modal logic as a logic of *frames*, and thus as a fragment of universal monadic second-order logic MSO.

This fragment, while generally not very expressive and missing many simple first-order properties, nevertheless penetrates deeply into MSO. Perhaps its most interesting features are the recursive axiomatisability of validity and its finite model property, together implying decidability – a rare phenomenon in second-order logic when considered over arbitrary structures rather than special ones.

In this section we present some classical results characterising modally definable classes of frames, and discuss how persistence of modal formulae with respect to various classes of general frames can be used to determine their model-theoretic properties.

8.1 Modal definability of frame properties

Here we address the question which classes of frames are definable by modal formulae. A classical result from [51] answers this question in a traditional model-theoretic fashion, albeit using a somewhat ad-hoc construction, called *SA-construction* ('state-of-affairs construction'). Algebraically, it corresponds to taking a subalgebra of a homomorphic image, thus allowing a 'translation' of Birkhoff's theorem in terms of frame constructions, and so characterising equational classes of algebras as those closed under subalgebras, homomorphic images and direct products (see Chapter 6 of this handbook). Theorem 117 gives a more natural characterisation of the modally definable *elementary* classes. Here is another definability-by-preservation result, due to van Benthem (see [126, Theorem 3.5], [127, Theorem 16.5], [129]).

THEOREM 115. *A class of frames $\mathcal{K}$ is modally definable by a set of $\mathcal{D}$-persistent formulae iff it is closed under generated subframes, bounded morphisms, disjoint unions and ultrafilter extensions, and reflects ultrafilter extensions.*

Proof. We already know from sections 2 and 6, and Proposition 114 that every $\mathcal{D}$-persistent formula satisfies all preservation conditions of the theorem, whence the easier direction. Conversely, let $\mathcal{K}$ satisfy the preservation conditions. We show that $\mathcal{K} = \mathrm{FR}(\mathrm{Th}_{\mathrm{ML}}(\mathcal{K}))$. Let $\mathfrak{F} \models \mathrm{Th}_{\mathrm{ML}}(\mathcal{K})$. Recall that $\mathfrak{F}_{\max}$ denotes the full general frame over the frame $\mathfrak{F}$. Using the duality theory between general frames and modal algebras, and Birkhoff's theorem, one can show that $\mathfrak{ue}(\mathfrak{F}_{\max})$ is isomorphic to a generated general subframe of a bounded morphic image of $\mathfrak{ue}(\mathfrak{G}_{\max})$ where $\mathfrak{G}$ is a disjoint union of frames from $\mathcal{K}$. Now, $\mathfrak{G} \in \mathcal{K}$; hence $\mathfrak{ue}(\mathfrak{G}) \in \mathcal{K}$. So, tracing the underlying frames and using the closure conditions, we eventually find that $\mathfrak{ue}(\mathfrak{F}) \in \mathcal{K}$, whence $\mathfrak{F} \in \mathcal{K}$. ❑

We note that checking the conditions of the theorem above, even in the case when the class of frames is first-order definable, may be a practically very difficult task. A testimony for that is the fact that preservation of first-order formulae under ultrafilter extensions is Π^1_1-hard [122, Thm 2.3.17].

In the rest of this section we compare the expressiveness of modal logic over frames with first-order logic and some of its extensions within monadic second-order logic.

8.2 Modal logic versus first-order logic on frames

We have already seen that modal languages are generally incomparable with first-order languages in terms of definability of frame properties. Indeed, while simple elementary properties, such as irreflexivity, escape the basic modal language, it can capture non-elementary properties such as the one defining the class of all transitive frames in which there are no infinite chains of successors. By a simple compactness argument, this class is not elementary, while it is well-known to be defined by the *Gödel–Löb formula* GL (see e.g. [75]). This example also shows that the compactness theorem with respect to frame validity fails in modal logic. The downward Löwenheim–Skolem–Tarski theorem fails here, too. E.g., McKinsey's formula $\Box\Diamond p \to \Diamond\Box p$ (see [127], or [5, p.133]) is valid in a certain uncountable frame, but not in any countable elementary subframe of it. Another important example of a non-elementary modal formula (in the extended setting with the star operation for transitive closures) is Segerberg's induction axiom [117] IND : $[\alpha^*](p \to [\alpha]p) \to (p \to [\alpha^*]p)$.

The model-theoretic interplay

We compare modal formulae (respectively, modally definable properties of frames) and first-order formulae (respectively, properties definable in $\mathrm{FO}(\tau)$) from two perspectives:

- Which modally definable frame properties are first-order definable?
- Which first-order properties of frames are modally definable?

As already mentioned, there are two natural notions of first-order definability: by means of single sentences and by means of theories (possibly infinite sets of sentences). Regarding modally definable classes, however, these turn out to be equivalent. Indeed, if the class of frames $\mathrm{FR}(\varphi)$ is the class of models of an infinite set of $\mathrm{FO}(\tau)$-formulae Γ, then $\Gamma \models \varphi$ with respect to frame validity, which is a Π^1_1-property. The compactness theorem of first-order logic applies here, and $\Gamma_0 \models \varphi$ for some finite $\Gamma_0 \subseteq \Gamma$. Hence $\mathrm{FR}(\varphi)$ is defined by the conjunction over Γ_0. We can therefore refer to modally definable

classes which are first-order definable, and to modal formulae defining such classes, as *elementary* without risk of confusion.

On the other hand, it seems to be still unknown whether there is any FO-sentence equivalent to an infinite set of basic modal formulae but not to a single formula.[14]

The validity preservation results from sections 2 and 6 imply that every modally definable class of frames $\mathrm{FR}(\varphi)$ is closed under generated subframes, bounded morphic images (in particular, isomorphic copies), and disjoint unions, while it reflects ultrafilter extensions and ultraproducts. If, moreover, the formula φ is elementary, then $\mathrm{FR}(\varphi)$ is closed under ultraproducts, too. Conversely, if $\mathrm{FR}(\varphi)$ is closed under ultraproducts then, by the Keisler–Shelah theorem, $\mathrm{FR}(\varphi)$ is elementary. Moreover, by Proposition 85, closure of $\mathrm{FR}(\varphi)$ under ultrapowers suffices, and therefore, closure under elementary equivalence in $\mathrm{FO}(\tau)$ suffices, too. The latter, in turn, characterises $\Sigma\Delta$-elementary classes, i.e., unions of elementary classes. Thus, we have the following model-theoretic characterisation of the elementary modal formulae (see [44, 47, 127]).

THEOREM 116. *For any modal formula φ the following are equivalent:*

(i) *φ is elementary.*

(ii) $\mathrm{FR}(\varphi)$ *is closed under ultraproducts.*

(iii) $\mathrm{FR}(\varphi)$ *is closed under ultrapowers.*

(iv) $\mathrm{FR}(\varphi)$ *is closed under elementary equivalence, i.e., $\Sigma\Delta$-elementary.*

The result above correspondingly characterises elementary classes of frames that are known to be modally definable. This raises the natural question how to characterise, in model theoretic terms, modal definability of an elementary class of frames. Again, a classical result from [51] answers that question. Here is a somewhat strengthened version (see [5, Theorem 5.54]).

THEOREM 117 (Goldblatt–Thomason). *If a class of frames $\mathcal{K}$ is closed under ultrapowers (in particular, if $\mathcal{K}$ is elementary), then $\mathcal{K}$ is modally definable iff it is closed under generated subframes, bounded morphisms, and disjoint unions, and reflects ultrafilter extensions.*

Proof. One direction is a direct application of the preservation results from sections 2 and 6. For the other direction note that, by Theorem 90 reduced to underlying frames, $\mathcal{K}$ is closed under ultrafilter extensions, too. Thus, Theorem 115 applies, so $\mathcal{K}$ is modally definable, moreover by a set of $\mathcal{D}$-persistent formulae. ❑

We end with an important related result, originally due to Fine [28], later strengthened and proved by van Benthem [127, Theorem 16.7] as a corollary to Theorem 115.[15]

We call a modal formula φ *complete* if the modal logic axiomatised by φ is complete for the class of frames defined by φ.

THEOREM 118 (Fine–van Benthem).
Every complete and elementary modal formula φ is $\mathcal{D}$-persistent.

Proof. $\mathrm{FR}(\varphi)$ satisfies all closure conditions of Theorem 115, so $\mathrm{FR}(\varphi) = \mathrm{FR}(\Gamma)$ for some set of $\mathcal{D}$-persistent formulae Γ. The modal logic $\mathbf{K}_\tau + \Gamma$, axiomatised with the set

[14]There are known cases, however, where a first-order definable property is infinitely, but not finitely, axiomatisable in some extended modal languages. See, e.g., [54].

[15]For a stronger algebraic version of this theorem see [45], [5, Theorem 5.56], or Chapter 6 of this handbook.

of axioms Γ, is canonical and therefore complete. Hence $\mathbf{K}_\tau + \Gamma \vdash \varphi$. By compactness of modal derivations, $\mathbf{K}_\tau + \Gamma_0 \vdash \varphi$ for some finite subset Γ_0 of Γ. By completeness of φ, all formulae from Γ_0 are theorems of $\mathbf{K}_\tau + \varphi$. Hence φ is axiomatically equivalent, and therefore frame-equivalent, too, to the conjunction of Γ_0, which is itself a $\mathcal{D}$-persistent formula. ❑

It is known ([28], see also section 8.2) that the converse to the above theorem does not hold, viz. not every $\mathcal{D}$-persistent formula is elementary. An example is $\Diamond\Box(p \vee q) \to \Diamond(\Box p \vee \Box q)$ (see [28]). Nor is every elementary modal formula $\mathcal{D}$-persistent, as there are incomplete elementary modal formulae (e.g., van Benthem's formula **vB** discussed in section 7.3, see [128, p.72], also in [5, p.216]).

It had been a longstanding open problem, posed by Fine, whether every modal logic axiomatised by $\mathcal{D}$-persistent formulae is complete with respect to *some* elementary class. This question has recently been answered negatively in [50].

Persistence and first-order definability

Some persistence properties of modal formulae imply that they are elementary. Perhaps the first interesting result in that vein is due to Lachlan [89] who proved that every $\mathcal{R}$-persistent formula is elementary. A strengthening of Lachlan's result, using the argument in Goldblatt's proof of it in [44], is that every (locally) $\mathcal{DI}$-persistent formula is (locally) elementary. First, note that local non-validity of a modal formula, being a Σ^1_1-property, is preserved by ultraproducts [12, Corollary 4.1.14]. By the Keisler–Shelah theorem it suffices to show that local validity of locally $\mathcal{DI}$-persistent formulae is preserved under ultraproducts. This follows from the fact that local validity of modal formulae is locally preserved in ultraproducts of general frames (Proposition 109), and that any ultraproduct of full general frames is a discrete general frame.

Let us now turn to $\mathcal{E}$-persistent formulae. They were first studied by van Benthem in [127] in connection with the *substitution method* which can be used to establish the first-order definability of Sahlqvist formulae (see section 8.2). The idea of the substitution method is to identify finitely many 'characteristic' first-order definable valuations of the variables occurring in a given formula, such that the formula is (locally) valid in every frame in which it is (locally) valid for those characteristic valuations. For all Sahlqvist formulae, just one such valuation, the *minimal* one amongst all those satisfying the antecedent of the formula, suffices. Van Benthem provided an alternative characterisation of locally and globally $\mathcal{E}$-persistent formulae, which implies that they are locally elementary.

Given a $\mathrm{FO}(\tau_\Phi)$-formula $\beta(\mathbf{x})$ with unary predicates $P_1, \ldots, P_n$, assuming that the variables $\mathbf{x}$ do not occur bound in β and the variables $z_1, \ldots, z_k, y$ do not occur in β at all, we define a *universally parameterised* $\mathrm{FO}(\tau)$-*substitution instance* of β to be any $\mathrm{FO}(\tau)$-formula $\forall z_1 \ldots \forall z_k \beta[\sigma_1/P_1, \ldots, \sigma_n/P_n]$ obtained from β by selecting $\mathrm{FO}(\tau)$-formulae $\sigma_i = \sigma_i(\mathbf{x}, z_1, \ldots, z_k, y)$ for $i = 1, \ldots, n$, uniformly substituting $\sigma_i[x/y]$ for every occurrence of $P_i x$, and then universally quantifying over $z_1, \ldots, z_k$. Let $\Theta(\beta)$ be the set of all universally parameterised $\mathrm{FO}(\tau)$-substitution instances of β.

DEFINITION 119. A modal formula $\varphi = \varphi(p_1, \ldots, p_n)$ is a *van Benthem formula* if $\Theta(\mathrm{ST}(\varphi; x_0)) \models \forall P_1 \ldots \forall P_n \mathrm{ST}(\varphi; x_0)$. We let VB denote the class of van Benthem formulae (defined slightly differently in [127] as the class M^1_{sub}).

THEOREM 120. *A modal formula is locally $\mathcal{E}$-persistent iff it is a van Benthem formula.*

Proof. Recall that $\mathfrak{E}(\mathfrak{F})$ is the minimal elementary general frame over the Kripke frame $\mathfrak{F}$. Let $\varphi(p_1, \ldots, p_n) \in \mathrm{VB}$ and suppose $\mathfrak{E}(\mathfrak{F}), w \models \varphi$ for some frame $\mathfrak{F}$. Take any universally parametrised $\mathrm{FO}(\tau)$-substitution instance $\forall z_1 \ldots \forall z_k \mathrm{ST}(\varphi)[\sigma_1/P_1, \ldots, \sigma_n/P_n]$. Let $w_1, \ldots, w_k \in \mathrm{dom}(\mathfrak{F})$ and $X_i := \{u \in \mathrm{dom}(\mathfrak{F}) \mid \mathfrak{F} \models \sigma_i(w, w_1, \ldots, w_k, u)\}$ for $i = 1, \ldots, n$. Since $X_1, \ldots, X_n$ are admissible in $\mathfrak{E}(\mathbf{F})$, $(\mathfrak{F}; X_1, \ldots, X_n; w) \models \mathrm{ST}(\varphi)(P_1, \ldots, P_n; x_0)$. Therefore, $\mathfrak{F}, w \models \forall z_1 \ldots \forall z_k \mathrm{ST}(\varphi)[\sigma_1/P_1, \ldots, \sigma_n/P_n]$. Since $\varphi \in \mathrm{VB}$, that implies $\mathfrak{F}, w \models \varphi$.

Conversely, let φ be locally $\mathcal{E}$-persistent and suppose $\mathfrak{F}, w \models \Theta(\mathrm{ST}(\varphi; x_0))$. Then, reversing the argument above, we find that $\mathfrak{E}(\mathfrak{F}), w \models \varphi$, and therefore $\mathfrak{F}, w \models \varphi$ by local $\mathcal{E}$-persistence of φ. ❑

We can now strengthen the earlier persistence-implies-elementary results.

THEOREM 121. *Every (locally) $\mathcal{E}$-persistent formula is (locally) elementary.*

Proof. Clearly, for every modal formula φ, $\forall P_1 \ldots \forall P_n \mathrm{ST}(\varphi; x_0) \models \Theta(\mathrm{ST}(\varphi; x_0))$. By compactness, every van Benthem formula is a logical consequence of a finite subset of $\Theta(\mathrm{ST}(\varphi; x_0))$, and hence is equivalent to the conjunction over that set. ❑

Consequently, not every $\mathcal{D}$-persistent formula is $\mathcal{E}$-persistent. Neither is every (locally) elementary modal formula (locally) $\mathcal{E}$-persistent. An example (see [127]) is the formula $\mathrm{Mk4} = (\Box p \to \Box\Box p) \wedge (\Box\Diamond p \to \Diamond\Box p)$, which is elementary and valid in the general frame $\langle \mathbb{N}, <, \mathbb{W}\rangle$ where $\mathbb{W}$ is the set of all finite and co-finite subsets of $\mathbb{N}$, while if fails in $\langle \mathbb{N}, <\rangle$. Since $\mathbb{W}$ contains precisely all parametrically first-order definable sets in $\langle \mathbb{N}, <\rangle$, it is $\mathfrak{E}(\langle \mathbb{N}, <\rangle)$, so Mk4 is not $\mathcal{E}$-persistent. Similarly, $\mathrm{Mk4}' = (\Box p \to \Box\Box p) \wedge \Box(\Box p \to \Box\Box p) \wedge (\Box\Diamond p \to \Diamond\Box p)$ is locally elementary,[16] but not locally $\mathcal{E}$-persistent.

Sahlqvist formulae and inductive formulae

The model-theoretic results discussed above, however elegant, are usually not easy to apply, and are of no use to find the actual first-order formula corresponding to the modal formula. It is therefore natural to look for simpler and *effective* sufficient conditions for first-order definability of modal formulae. There can be no completely satisfactory outcome of that search, because that property is not decidable [11], and (at least) in a modal language with more than one modality, not even analytical [122, Thm 2.6.5]. Still, several increasingly general results to that aim were obtained during the 1970's, culminating with the celebrated *Sahlqvist theorem*, which not only identifies a large syntactic class of elementary modal formulae (see a simple definition of that class below), but also proves their canonicity. A variety of expositions of Sahlqvist's theorem can be found in several sources, e.g. [113, 115, 5, 84, 10], Chapters 6 and 7 of this handbook. Here we outline a generalisation of the class of Sahlqvist formulae in monadic poly-modal languages, sharing the same virtues as the original class, viz. the *inductive formulae* introduced and studied for arbitrary polyadic languages in [55].

We fix a modal language $\mathrm{ML}(\tau)$.

DEFINITION 122. Let # be a symbol not belonging to $\mathrm{ML}(\tau)$. Then a *box-form of #* in $\mathrm{ML}(\tau)$ is defined recursively as follows:

[16] The fact that Mk4 and Mk4′ are elementary is far from trivial, as the proof requires a form of the Axiom of Choice and cannot be formalised in ZF.

(i) $\#$ is a box-form of $\#$;

(ii) If $\mathbf{B}(\#)$ is a box-form of $\#$ and $\Box$ is a box-modality in $\mathrm{ML}(\tau)$, then $\Box\mathbf{B}(\#)$ is a box-form of $\#$;

(iii) If $\mathbf{B}(\#)$ is a box-form of $\#$ and A is a positive τ-formula, then $A \to \mathbf{B}(\#)$ is a box-form of $\#$.

Thus, box-forms of $\#$ are, up to semantic equivalence, of the type $\Box_1(A_1 \to \Box_2(A_2 \to \dots \Box_n(A_n \to \#)\dots)$, where $\Box_1, \dots, \Box_n$ are box-modalities and $A_1, \dots, A_n$ are positive formulae in $\mathrm{ML}(\tau)$.

DEFINITION 123. Given a propositional variable p, a *box-formula of* p is the result $\mathbf{B}(p)$ of substitution of p for $\#$ in any box-form $\mathbf{B}(\#)$. The last occurrence of the variable p is the *head* of $\mathbf{B}(p)$ and every other occurrence of a variable in $\mathbf{B}(p)$ is *inessential* there.

DEFINITION 124. A *(monadic) regular formula* is any modal formula built from positive formulae and negations of box-formulae by applying conjunctions, disjunctions, and boxes.

DEFINITION 125. The *dependency digraph* of a set $\mathcal{B} = \{\mathbf{B}_1(p_1), \dots, \mathbf{B}_n(p_n)\}$ of box-formulae is the digraph $G = \langle V, E\rangle$ where $V = \{p_1, \dots, p_n\}$ is the set of heads in A, and $p_i E p_j$ iff p_i occurs as an inessential variable in a box-formula from $\mathcal{B}$ with a head p_j. A digraph is called *acyclic* if it does not contain oriented cycles.

DEFINITION 126. An *inductive formula* is a regular formula with an acyclic dependency digraph of the set of all box-formulae occurring as subformulae in it.

We note that Sahlqvist formulae, up to semantic equivalence, are precisely those regular formulae in which the box-formulae are just *boxed atoms*, i.e., propositional variables prefixed by possibly empty strings of boxes. Thus, all Sahlqvist formulae fall into a simple particular case of inductive formulae, where the dependency digraph has no arcs at all.

The following extension of Sahlqvist's theorem was established in [55].

THEOREM 127. *Each inductive formula is locally elementary and locally $\mathcal{D}$-persistent. Moreover, its local first-order equivalent can be computed effectively.*

The inductive formulae are van Benthem formulae which, just like Sahlqvist formulae, have first-order definable minimal valuations, but they can only be computed inductively, in steps following the arcs of the dependency digraph, from sources to sinks.

Sahlqvist formulae satisfy a certain persistence property which can be extracted from the syntactic shape of the first-order formulae defining their minimal valuations. In the basic modal language these valuations are either the empty set, or the whole domain, or are finite unions of sets of the type $R^n(y)$ (recall that R^n is the n-fold composition of R with itself). Following [55], let us call a general frame *ample* if it contains all such sets as admissible, and the modal formulae locally persistent with respect to all ample general frames, *locally $\mathcal{A}$-persistent.* Thus, all Sahlqvist formulae in $\mathrm{ML}(\Diamond)$ are locally $\mathcal{A}$-persistent, and this property enables us to show that a given formula is not (even semantically equivalent to) a Sahlqvist formula.

EXAMPLE 128. As proved in [55], the formula $D = p \wedge \Box(\Diamond p \to \Box q) \to \Diamond\Box\Box q$ is not $\mathcal{A}$-persistent, and hence not equivalent to any Sahlqvist formula in $\mathrm{ML}(\Diamond)$. However, it is an inductive formula, whose dependency digraph over the set of heads $\{p, q\}$ has only one edge, from p to q. It has a local first-order correspondent $\mathrm{FO}(D) = \exists y(Rxy \wedge \forall z(R^2yz \to$

$\exists u(Rxu \wedge Rux \wedge Ruz)))$, which is not equivalent to a Kracht formula (i.e., a first-order equivalent to a Sahlqvist formula, see [84]).

The class of inductive formulae does not exhaust the potential of the method of substitutions, (in particular, minimal valuations), since, being syntactically defined (like Sahlqvist formulae), it is not closed even under tautological equivalence.

A more general and robust *algorithmic* approach to identifying elementary and $\mathcal{D}$-persistent modal formulae (covering all inductive formulae) is outlined in [14]. The algorithm presented there is based on a modal version of Ackermann's lemma (which essentially formalises the idea of minimal valuations) and, when successful, computes effectively a first-order equivalent of the input modal formula and at the same time establishes its $\mathcal{D}$-persistence.

Shallow formulae and $\mathcal{R}$-persistence

The property of $\mathcal{R}$-persistence is much stronger than $\mathcal{D}$-persistence. Perhaps the largest syntactic class of $\mathcal{R}$-persistent formulae identified so far is the class of *shallow formulae* [122, Thm 2.4.7]: those in which every occurrence of a propositional variable is in the scope of at most one modal operator. Note that syntactically shallow formulae are not subsumed by the class of Sahlqvist formulae, nor even by the class of inductive formulae.

8.3 Modal logic and first-order logic with least fixed points

With every first-order language FO(τ) we associate its extension LFP(τ) with *least fixed point operators*. For background on LFP see e.g. [25] or [2]. LFP is a rather expressive proper extension of FO which however still shares nice properties with with FO, e.g., the downward Löwenheim–Skolem theorem [30] and the 0-1 law (see [64]).

Which modal formulae are (locally) definable in LFP(τ)? Which LFP(τ)-formulae are modally definable on frames? No explicit model-theoretic criteria seem to be known as yet and these questions are most likely undecidable.

A number of well-known non-elementary modal formulae, such as the Gödel-Löb formula GL and Segerberg's induction axiom IND have local equivalents in LFP(τ) while, for instance, the McKinsey formula is outside that class. Indeed, take van Benthem's uncountable frame from [127] in which that formula is valid. Flum's argument from [30], proving the downward Löwenheim–Skolem–Tarski theorem for LFP, produces a countable elementary subframe of it which must satisfy that formula, too, which is not possible, as shown in [127].

Still, a large, effectively defined class of LFP(τ)-expressible modal formulae can be identified by noting that the idea of using minimal valuations to eliminate the universal second-order quantifiers in the standard translation of frame validity of modal formulae goes beyond first-order logic. Indeed, the same idea works perfectly for all (polyadic) regular formulae, defined for monadic languages in section 8.2. In cases where the dependency graph has loops and cycles, the minimal valuations are recursively defined and eventually expressed in LFP(τ). In particular, this applies to Gödel–Löb and Segerberg formulae, being regular formulae. The following was shown in [55].

THEOREM 129. *Every regular formula has a local correspondent in* LFP(τ)*, which can be obtained effectively.*

We illustrate the idea of computing LFP(τ)-equivalents of regular formulae with GL.

$$\mathrm{ST}(\mathrm{GL}) = \forall x_1(x_0Rx_1 \to (\forall x_0(x_1Rx_0 \to Px_0) \to Px_1)) \to \forall x_1(x_0Rx_1 \to Px_1),$$

which can be rewritten as $\forall x_1(x_0Rx_1 \to (R[x_1] \subseteq P) \to Px_1)) \to R[x_0] \subseteq P$ (where $R[x] := \{y \mid xRy\}$). The antecedent can be expressed as

$$\Phi(P) \subseteq P, \quad \text{where} \quad \Phi(P) = \{x_1 \mid x_0Rx_1 \wedge R[x_1] \subseteq P\}\,.$$

Note that, since $\Phi(P)$ is positive in P, and hence monotone, there is a $\subseteq$-minimal valuation for P satisfying $\Phi(P) \subseteq P$, viz. $V_m(p) = \mu X.\Phi(X)$. Then, the local equivalent of GL in LFP(τ) is obtained by substituting that minimal valuation in the consequent: $\mathrm{LFP}(\tau)(GL; x_0) = \forall x_1(x_0Rx_1 \to \mu X.\Phi(X)(x_1))$. By unfolding, based on the Knaster–Tarski theorem, that equivalent is:

$\forall x_1(x_0Rx_1 \to$

$\exists n \geq 0\ \forall y_1 \ldots \forall y_n(x_1Ry_1 \to x_0Ry_1 \wedge (\ldots(y_{n-1}Ry_n \to x_0Ry_n \wedge R[y_n] = \emptyset)\ldots)),$

i.e., 'local' transitivity and non-existence of infinite R-chains starting at x_0.

While Theorem 129 may be regarded as an extension of the definability part of the Sahlqvist theorem, it cannot match the canonicity part of it. Not only are there regular formulae which are not $\mathcal{D}$-persistent (e.g., GL and IND) but there are even ones which are not complete, such as $\Box(\Box p \leftrightarrow p) \to \Box p$ from [6], which can be easily pre-processed into a semantically equivalent regular formula. It is weaker than GL but has the same class of frames, and is therefore incomplete. On the other hand, it is a plausible conjecture that every modal formula with a minimal valuation expressible in LFP(τ) is semantically equivalent to a regular formula.

In order to apply the method of minimal valuations, one has to identify, en route, those FO(τ)-formulae γ for which there is a minimal interpretation for each occurring unary predicate P. In recent work van Benthem [132] has obtained syntactic and model theoretic characterisations of these formulae, involving predicates of arbitrary arity (see Chapter 1 of this handbook).

Finally, we note that an algorithm for computing LFP(τ)-equivalents of classical modal formulae, based on Ackermann's method for second-order quantifier elimination, and in particular covering the example above, has been developed in [103].

8.4 Modal logic and second-order logic

The standard translation embeds ML(τ), with respect to frame validity, into the monadic Π^1_1-extension of the first-order language FO(τ_Φ). We already know that the embedding is proper. Still, a natural question arises whether the preservation conditions of Theorem 117 are sufficient to guarantee modal definability of monadic Π^1_1-formulae, as well. As van Benthem has noted in [127, p.53], this is not the case in the basic modal language, witnessed by the property 'non-existence of infinite R-chains' (i.e., well-foundedness of R^{-1}), which satisfies all those preservation conditions and moreover is bisimulation invariant. Still, that property of frames is defined in the extension of the basic modal language with the universal modality $[U]$, by the formula $[U](\Box p \to p) \to p$ (see [54]). (Contrast this with Observation 42, that as a property of Kripke structures, it is not definable even in ML_∞.) Thus, one may ask if the natural preservation conditions characterising modal definability of elementary properties (closure under generated subframes, bounded morphisms, and disjoint unions, and reflection of ultrafilter extensions) do not apply also to a

wider class (if not the whole of Π^1_1), but for a suitably extended modal language? Surely, some of the results characterising modal definability of properties of Kripke structures would still be useful and relevant here: if the first-order matrix of a Π^1_1-formula, where all second order quantifiers are in a prefix, meets the conditions for having a modal correspondent on Kripke structures, then the whole formula is frame-definable by the same modal correspondent. It is not currently known if this observation can be turned into a general criterion for modal definability of monadic second-order formulae.

Modal logic penetrates quite deep into monadic second-order logic $\mathrm{MSO}(R)$ (with full quantification over unary predicate variables, over the vocabulary with the single binary relation R). As proved by Thomason [124], logical consequence in terms of frame validity of the latter can be reduced to the former in the following sense. There exists an effective translation t of $\mathrm{MSO}(R)$ into ML, and a special modal formula δ such that for every set Σ of $\mathrm{MSO}(R)$-sentences and any $\mathrm{MSO}(R)$-sentence φ: $\Sigma \models_2 \varphi$ iff $\{\delta\} \cup t[\Sigma] \models_{\mathrm{FR}} t(\varphi)$. Here $\models_2$ denotes second-order semantic consequence which, as a consequence from Tarski's non-definability theorem, is not arithmetically definable, and $\Gamma \vDash_{\mathrm{FR}} \psi$ means that the modal formula ψ is valid in every frame where all modal formulae from Γ are valid. Consequently, $\vDash_{\mathrm{FR}}$ is not recursively axiomatisable, unlike validity in modal logic.

Furthermore, as noted in [127, p.23], full second-order logic, and even the theory of finite types, can be reduced to $\mathrm{MSO}(R)$, too.

For more on the relations between modal logic and second-order logic, see [127], [24], and Chapter 10 of this handbook. Also, [122, 121, Chapter 12] considers the extension of modal logic with propositional quantifiers, which goes much farther into second-order logic.

9 FINITE MODEL THEORY OF MODAL LOGICS

9.1 Finite versus classical model theory

When only finite structures are admitted, the model theoretic basis changes dramatically. For instance, unless the logic under consideration has the finite model property, satisfiability does not imply finite satisfiability, and hence a semantic consequence $\varphi \models \psi$ may be true in the sense of finite models without being classically valid. Crucial tools of classical model theory, most notably the completeness and compactness theorems for FO, fail in restriction to just finite models. From a modelling point of view, on the other hand, the restriction to just finite models is often natural. In applications, in which the intended models ought to be finite, reasoning on the basis also of infinite models may be inadequate and give misleading results. Applications in computer science like specification and verification, or also database theory, for instance, often call for the restriction to finite models, and have had a significant impact on the development of finite model theory.

The methodological shift encountered is highlighted by the failure of classical theorems and tools, most notably of the compactness theorem but also most other key theorems from classical model theory in its wake, see [25]. Certainly results from classical model theory cannot be expected to go through automatically; often they fail, and some still obtain, albeit with new proofs. Modal model theory, in particular, has a number of examples of the latter kind, and sometimes the new proofs shed new light also on the classical version. For some concrete examples, close to (classical) modal model theory, which

illustrate the interesting relationship with finite model theory, consider the following.

Interpolation for ML goes through via the finite model property (FMP), treated in section 3.3. If $\models \varphi \to \psi$ is valid in finite structures, it must also be valid generally, as a counterexample $\mathfrak{M}, w \models \varphi \wedge \neg\psi$ would also yield a counterexample in the sense of finite model theory, by FMP. Clearly a classical interpolant χ with $\models (\varphi \to \chi) \wedge (\chi \to \psi)$, is an interpolant also in the sense of finite model theory.

The modal characterisation theorem. Note how both sides of the equivalence expressed in Theorem 55 change their meaning when interpreted in the sense of finite model theory: both bisimulation invariance and logical equivalence only refer to finite structures. In particular, bisimulation invariance in finite structures does not imply bisimulation invariance over all structures. Trivial examples are provided by formulae without finite models that happen not to be bisimulation invariant for infinite models. Also, while Ehrenfeucht–Fraïssé techniques remain valid, compactness does not and the classical proof with its necessary detour through infinite models is no longer available. As discussed in section 4.2, however, the theorem itself persists in the form of Theorem 61 as a theorem of finite model theory due to Rosen [112]. Interestingly the new proofs in [112, 105] are valid classically as well as in finite model theory and have lead to additional insights into the classical result. In contrast, the failure of the corresponding characterisation theorem for FO^2 in finite model theory shows that the finite model property does not guarantee a smooth passage to finite model theory. While an FO sentence that is (classically) invariant under 2-pebble game equivalence is logically equivalent to a sentence in FO^2, this characterisation breaks down for finite model theory. The FO sentence saying that a binary relation is a linear ordering, which is 2-pebble invariant only in restriction to finite structures, is not expressible in FO^2 even over finite structures.

Similarly, Rosen [112] has a proof of the finite model theory version of the modal existential preservation theorem: $\varphi \in \mathrm{ML}$ is preserved under extensions (holds inside the whole Kripke structure if it holds in a substructure) iff it is equivalent to an existential modal formula (built from positive and negated atoms by means of only $\wedge$, $\vee$ and $\Diamond$ – disallowing $\Box$ or nesting of $\neg$ and $\Diamond$). The corresponding preservation theorem for first-order logic is known to become invalid in restriction to just finite structures.

Modal logic stands out in comparison with first-order logic or the FO^k in having a comparatively smooth finite model theory that preserves a number of classical theorems, as is the case for the above examples.

The variations of basic modal logic mentioned in section 5.1 have partly also been investigated with respect to their finite model theory, with several results that suggest a similarly smooth behaviour. Their characterisations as fragments of FO, in terms of invariance under correspondingly refined notions of bisimulation, have been studied in finite model theory in [106] with further ramifications w.r.t. other restricted classes of finite frames in [17]. Just as is the case with van Benthem–Rosen characterisation, Theorems 55 and 61 surprisingly many of these characterisations go through in restriction to finite Kripke structures just as classically, albeit with rather specific new proofs. The following may serve as a typical representative for several related results from [106, 17]. Also compare Proposition 68; this should be contrasted with the failure of, for instance, the corresponding characterisation of FO^2 in finite model theory.

THEOREM 130. *For any $\varphi(x) \in \mathrm{FO}$, the following are equivalent:*

(i) *φ is invariant under global bisimulation over finite Kripke structures.*

(ii) *φ is equivalent to a formula of* ML[∀] *over finite Kripke structures.*

Similarly are equivalent:

(i) *φ is bisimulation invariant over finite, rooted Kripke structures.*

(ii) *φ is equivalent to a formula of* ML[∀] *over finite, rooted Kripke structures.*

Related open problems concern the status in finite model theory of Theorem 65, for the guarded fragment GF in arbitrary relational similarity types, and particularly strikingly of Theorem 76, for the modal μ-calculus.

But finite model theory also deals with new questions, which only arise in the context of finite structures. We devote the rest of this section to two sketches dealing with two very specific issues of this kind: one from descriptive complexity (section 9.2), the other one 0-1 laws (section 9.3). Descriptive complexity deals with the relationship between the algorithmic complexity and the logical definability of properties of finite structures; here finite structures feature as input to algorithmic problems and logic becomes a measure of complexity. In 0-1 laws, and more generally asymptotic probability, one deals with the statistics of logically defined properties over the collection of all size n structures in the limit as n goes to infinity; here finite structures form the sample space for probabilistic analysis. Compare [25, 93] for general background on these topics in finite model theory.

9.2 Capturing bisimulation invariant Ptime

Descriptive complexity aims for the description and analysis of computational complexity by means of logics. A key example is the long open problem of a *logic for Ptime*. One seeks a logic (with effective syntax) whose formulae define precisely those classes of finite relational structures, for which membership can be decided in polynomial time.[17] By a well-known result of Immerman [76] and Vardi [134], the least fixed point extension of first-order logic, LFP, is the solution for classes of finite, linearly ordered relational structures. The problem remains open to date for not necessarily ordered structures. Interestingly, the corresponding problem for bisimulation closed classes of finite Kripke structures does admit a natural solution [104] (cf. [94] for another, related capturing result).

Consider the framework of basic modal logic with a single modality associated with the binary relation R and with finitely many atomic propositions p_i. Let $\mathcal{Q}$ be a class of finite pointed Kripke structures (i.e., a property of finite pointed Kripke structures) of that type. $\mathcal{Q}$ corresponds to a *bisimulation invariant* property if it is closed under bisimulation in the sense that for any two $(\mathfrak{M}, u) \rightleftarrows (\mathfrak{M}', u')$: $(\mathfrak{M}, u) \in \mathcal{Q}$ iff $(\mathfrak{M}', u') \in \mathcal{Q}$. Recall the bisimulation quotients $\mathfrak{M}[u]/\rho^{\mathfrak{M}}$ of pointed Kripke structures $(\mathfrak{M}, u)$ as discussed in section 3.6. Bisimulation closure of $\mathcal{Q}$ implies that

$$\mathcal{Q} = \big\{(\mathfrak{M}, u) \mid (\mathfrak{M}[u]/\rho^{\mathfrak{M}}, [u]_{\rho^{\mathfrak{M}}}) \in \mathcal{Q}\big\}.$$

Membership in $\mathcal{Q}$ can therefore be determined via passage to canonical quotient representations, and in terms of the intersection of $\mathcal{Q}$ with the class $\mathcal{C}$ of all canonical quotient representations. Note that $\mathcal{C}$ consists of all finite rooted Kripke structures of the appropriate type in which each bisimulation type is realised exactly once (in other words, with

[17] One also has to require an effective link from syntax to Ptime algorithms for its evaluation, in order to avoid pathological solutions.

identity as the largest bisimulation). As largest bisimulations and bisimulation quotients are polynomial time computable, it follows that $\mathcal{Q}$ is in Ptime if, and only if, $\mathcal{Q} \cap \mathcal{C}$ is. The following special property of $\mathcal{C}$ opens up a reduction to the case of linearly ordered structures, which then leads to the desired capturing result. By a canonical linear ordering of a structure we mean an ordering that is determined by the isomorphism type of that structure.

LEMMA 131. *There is a polynomial time algorithm which for every* $(\mathfrak{M}, u) \in \mathcal{C}$ *computes a canonical linear ordering of the domain.*

In fact, a linear ordering w.r.t. bisimulation type can be generated in an inductive refinement procedure which, in its n-th stage, produces a linear ordering of the $\rightleftarrows_n$-classes within any given finite Kripke structure. This is based on a lexicographic lift of the ordering on $\rightleftarrows_n$-classes to an ordering of the $\rightleftarrows_{n+1}$-classes, similar to the colour refinement technique in graph theory. Over any finite Kripke structure the common refinement of this process is a linear ordering of $\rightleftarrows$-classes; for structures in $\mathcal{C}$ one obtains an actual linear ordering, as each $\rightleftarrows$-class is inhabited by a single state.

Moreover, a representation of this linearly ordered version of the quotient structure $\mathfrak{M}[u]/\rho^{\mathfrak{M}}$ is uniformly LFP-definable over the given structures $(\mathfrak{M}, u)$ themselves. This means that in LFP over the $(\mathfrak{M}, u)$ one can also uniformly define any LFP definable property of their linearly ordered quotients $\mathfrak{M}[u]/\rho^{\mathfrak{M}}$. By the Immerman–Vardi result this includes all Ptime properties of these quotient structures, since they are linearly ordered. Together these observations yield an abstract capturing result: an effective syntactic normal form for the definition of precisely those bisimulation invariant properties that are in Ptime. As shown in [104] one can further isolate a natural extension of the modal μ-calculus, a multi-dimensional μ-calculus $\mathrm{L}^{\omega}_{\mu}$, with the property that a class $\mathcal{Q}$ of finite pointed Kripke structures is bisimulation closed and in Ptime if, and only if, $\mathcal{Q}$ is the class of finite models of a formula $\varphi \in \mathrm{L}^{\omega}_{\mu}$. The logic $\mathrm{L}^{\omega}_{\mu}$ is the natural bisimulation-safe least fixed-point extension of basic modal logic over the n-th cartesian power of a Kripke structure (intuitively: n-dimensional ML), for arbitrary $n \in \mathbb{N}$.

PROPOSITION 132. *Let* $\mathcal{Q}$ *be a class of finite pointed Kripke structures of fixed finite type. Then the following are equivalent:*

(i) $\mathcal{Q}$ *is bisimulation closed and in Ptime.*

(ii) $\mathcal{Q}$ *is definable by a formula of the multi-dimensional* μ*-calculus* $\mathrm{L}^{\omega}_{\mu}$.

9.3 0-1 laws in modal logic

Another of the major specific topics in finite model theory is the asymptotic behaviour of the probability for a given property $\mathcal{P}$ to be true in a randomly chosen structure of size n (taken up to isomorphism), in a suitably defined probabilistic space. If that probability has a limit as n increases without bound, that limit is called the *(unlabelled) asymptotic probability of* $\mathcal{P}$.

A fundamental result in this area is the *0-1 law* for first-order logic, stating that the asymptotic probability for every first-order definable property of relational structures exists and equals either 0 or 1, i.e., every such property is either *almost surely true* or *almost surely false*. This result was first proved in [40] (using 'almost sure' quantifier elimination), later established independently by Fagin [27] who moreover obtained a purely logical characterisation of the set of first-order sentences that are almost surely

true, as the first-order theory of the so-called *countable random structure*. Prior to Fagin's discovery, Gaifman had studied in [37] infinite random structures as probabilistic models for arbitrary relational first-order languages and had proved that the first-order theory of such structures is axiomatised by an infinite set of *extension axioms*: sentences that require every n-tuple to be extendible to an $(n+1)$-tuple in every possible (i.e., consistent) way. Furthermore, he showed that the first-order theory of all extension axioms is complete and ω-categorical.[18] Thus, Fagin established the following *transfer theorem*, which immediately implies the 0-1 law: a first-order property of relational structures is almost surely true iff it is true in the (unique, up to isomorphism) countable random structure. Grandjean [63] proved that the complexity of checking if a given first-order formula is almost surely true is decidable in Pspace, in sharp contrast to Trachtenbrot's theorem that validity of first-order formulae on *all* finite structures is not even recursively axiomatisable.

The transfer theorem was subsequently extended and the 0-1 law proved for several extensions of first-order logic: for first-order logic with fixed point operators by Blass, Gurevich and Kozen, later subsumed by the 0-1 law for infinitary logic with finitely many variables $L^{\omega}_{\infty\omega}$, proved by Kolaitis and Vardi; for some prefix-defined fragments of monadic second-order logic, again by Kolaitis and Vardi, who also established curious parallel between decidability and 0-1 laws for such fragments. On the other hand, the 0-1 law fails in monadic second-order logic, even in its Σ^1_1-fragment. For references and further details on these results, see, e.g., [64, 25, 93, 53].

In the framework of modal logic, there are two natural notions of (asymptotic) probability 'in the finite': with respect to Kripke structures and with respect to frames. The 0-1 law with respect to Kripke structures follows directly from Fagin's theorem. Moreover, Halpern and Kapron [66] showed that the modal formulae almost surely valid in finite Kripke structures are precisely the theorems of the non-normal *Carnap's logic* [8]. As for almost sure frame validity, a complete axiomatisation of the modal logic ML^r of the countable random frame has been obtained in [53], where it has also been proved that ML^r has the finite model property and is decidable. It is also shown there that not all modal formulae that are almost surely frame-valid are in ML^r, thus refuting the transfer theorem for frame validity in modal logic. Perhaps the simplest such formula, which fails in the countable random frame, is $\neg\Box\Box(p \leftrightarrow \neg\Diamond p)$, proven later in [90] to be almost surely true. Note that no such formula is frame-definable in fixed point logic LFP, or even in $L^{\omega}_{\infty\omega}$, because the transfer theorem does hold for these.

The failure of the transfer theorem for frame validity in modal logic cast a serious doubt on the truth of the 0-1 law there (claimed in [66]) which was soon justified by le Bars [90] who proved that the formula $\neg p \land q \land \Box\Box((p \lor q) \to \neg\Diamond(p \lor q)) \to \Diamond\Box\neg p$ has no asymptotic probability, by using involved combinatorial-probabilistic methods. Thus, basic modal logic provides the smallest currently known natural fragment of monadic Π^1_1 (resp. Σ^1_1), in a vocabulary with just a single binary relation, where the 0-1 law fails.

As noted in [53] the modal formulae which are almost surely frame-valid form a normal modal logic ML^{as}, which contains ML^r. It is a currently open problem whether ML^{as} is decidable, and its complete axiomatisation has not been established yet. However, a conjecture raised in [53] claims that all axioms that have to be added to ML^r

[18] The probabilistic aspect of this result is rather curious: it means that, assuming uniform distribution, any randomly constructed countable relational structure is isomorphic with probability 1 to the countable random structure! In the case of graphs, that structure was previously known as the *Radó graph.*

in order to axiomatise ML^{as} are of a uniform, semantic nature, namely: there is an infinite collection $\mathcal{F}$ of special finite frames, and each $\mathfrak{F} \in \mathcal{F}$ determines an axiom $\varphi_{\mathfrak{F}}$ valid in 'almost every' finite frame[19] iff that frame cannot be mapped by a bounded morphism onto $\mathfrak{F}$. For instance, the formula $\neg\Box\Box(p \leftrightarrow \neg\Diamond p)$ corresponds to the frame $\langle\{a,b\},\{\langle a,b\rangle,\langle b,a\rangle,\langle b,b\rangle\}\rangle$.

CONCLUDING REMARKS

In summary, the semantics of modal logic has (at least) two emblematic features which have a crucial impact on its model theory and which we have attempted to reflect in the composition of this chapter.

Modal logic is local. Truth of a formula is evaluated at a current state (possible world); this localisation is preserved (and carried) along the edges of the accessibility relations by the restricted, relativised quantification corresponding to the modal operators. This feature is reflected by the notion of *bisimulation* between states and between Kripke structures, respectively. The notion of bisimulation invariance plays a key role in characterising what is modally definable, as captured in the van Benthem–Rosen theorem (Theorems 55 and 61 here). Moreover, bisimulation (and its game characterisation) plays a role in modal model theory analogous to that of partial isomorphism (and its Ehrenfeucht–Fraïssé characterisation) in classical model theory. From yet another perspective, the characteristic power of preservation under bisimulations in modal logic can be compared to the characteristic power of preservation under ultraproducts in first-order logic. Quite naturally, therefore, bisimulation emerges as the central and unifying truth-preserving model-theoretic construction in modal logic, and all other basic constructions on which the classical model theory of modal logic builds (generated substructures, bounded morphisms, disjoint unions) are definable in terms of it or at least closely related to it. By systematically developing the bisimulation-based approach to modal model theory in this chapter, we hope to have given a modern treatment on this classical theme. Furthermore, the central role of bisimulations and bisimulation invariance properties is so robustly preserved, *mutatis mutandis*, in the rich and diverse variety of extensions of basic modal logic, that it can be adopted as a benchmark of what constitutes a modal language.

Modal logic is multi-layered. On Kripke structures the modal language is a bounded variable, guarded fragment of first-order logic, while on Kripke frames, due to universal quantification over valuations, it becomes a fragment of universal monadic second-order logic. Each of these semantic layers leads to its own model-theoretic agenda and development, but the two interact closely through various model-theoretic constructions and preservation results presented here, and blend together in the notion of general frames, dually re-incarnated as modal algebras. General frames emerge as a third, intermediate semantic layer of modal logic, casting a bridge between the other two. In particular, by means of a hierarchy of persistency properties, general frames provide a yardstick to measure the 'expressive complexity' of modal formulae, and determine their model-theoretic

[19] More precisely, in every finite frame in which each state is reachable from any other state by a path of length ≤ 2.

behaviour. This chapter presents the basics of the modal model theory in each of these three layers and illustrates the use of the main tools and results arising in each one of them.

While trying to give a comprehensive account of the main issues and results of both classical and modern model theory of modal logic, we have not covered a number of important and relevant topics and research developments, either for lack of space or because they are adequately treated in other chapters of this handbook. A certainly incomplete list of the more conspicuous omissions (in no particular order) includes:

- model theory of extended modal languages: see [18] and [122] for a recent treatise;
- model theory of combined modal logics: see Chapter 15 of this handbook and [35];
- Lindström-type theorems for modal logic: see [19, 133];
- reductions of polyadic to monadic modal languages and their model theoretic implications, including transfer of properties: see [85, 41], and Chapter 8 of this handbook;
- Kracht's internal definability theory [84];
- Zakharyaschev's canonical formulae, providing a uniform characterisation of normal modal logics extending K4: see [10, 9], and Chapter 7 of this handbook;
- model-building techniques such as mosaics and networks used for more advanced completeness and decidability proofs: see, e.g., [99] and [5, Ch. 6.4 and 7.4].
- model completions in modal logic [39];
- bisimulation quantifiers and their use for proving uniform interpolation of various modal logics by Visser [139], Ghilardi and Zawadowski [39] (where bisimulation quantifiers are related to model completions), and of the modal mu-calculus by D'Agostino and Hollenberg [16].

It is natural to conclude a handbook chapter by attempting to identify main general trends of the current and future development of the topic under consideration.

To begin with, let us recall and revisit van Benthem's three 'pillars of wisdom' supporting the classical edifice of modal logic: the *Definability (Correspondence), Completeness, and Duality theories* [128]. Each of these has played a crucial role in the development of modal model theory, and will continue to play such a role, with an accordingly modernised and updated agenda.

In particular, analysing the expressive power of modal languages with respect to each of its semantic layers remains one of the main directions of research in modal logic, of growing importance and complexity, due to the active expansion and diversification of modal logic. Accordingly, the classical correspondence theory between modal and first-order logic, much of which has been reflected in the chapter, is gradually ramifying into a hierarchy of correspondence theories, aiming at mapping the variety of modal logics into the hierarchy of classical logical languages centered around first-order logic. An example is the currently emerging correspondence theory between modal logic and LFP.

Establishing completeness results of modal deductive systems designed to capture an intended semantics also remains one of the core areas of modal logic (as of logic in general) which requires increasingly sophisticated and powerful techniques to match the more and more complex modal languages and their semantics. The involved completeness proofs for the modal mu-calculus (see Chapter 12 of this handbook) and CTL* (see [111]), and the still open completeness problem for Parikh's (full) Game Logic (see Chapter 20 of this handbook) are cases in point.

Likewise for decidability and complexity, where model-theoretic tools and techniques, such as the model-building techniques mentioned above as well as game-theoretic methods, are gaining increasing recognition and variety of applications.

New directions and problem areas in modal model theory itself, or using model-theoretic methods, are emerging, too. Many of them, such as finite model theory and descriptive complexity, finite and infinite state model checking, arise from actual or potential applications of modal logic to computer science and related fields and follow recent trends in classical model theory. Let us note, however, that while the present day model theory of modal logic is still using mainly results and techniques from the classical era of first-order model theory, the enormous development and sophistication of that field over the past decades is yet to make its full impact on modal model theory.

In closing, being aware that we cannot possibly offer a definitive treatment of such a rich and dynamic subject as the model theory of modal logic, we hope to have whetted readers' appetites and their desire to explore it further and to add to it new discoveries of their own.

ACKNOWLEDGEMENTS

We are indebted to Maarten de Rijke for his careful reading, critical remarks, corrections and numerous helpful suggestions for improvement. We also thank Achim Blumensath, Willem Conradie, Ian Hodkinson, Mark Reynolds, Balder ten Cate and Johan van Benthem for valuable comments or corrections. Most of the work of Valentin Goranko on this chapter was done while he worked at the Rand Afrikaans University (now, the University of Johannesburg), and he would like to acknowledge the financial support provided by that university and by the National Research Foundation of South Africa. Martin Otto was partially supported by a British EPSRC grant, originally held at the University of Wales Swansea, during a major period of his work on this chapter.

BIBLIOGRAPHY

[1] H. Andreka, J. van Benthem, and I. Nemeti. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217–274, 1998.

[2] A. Arnold and D. Niwinski. *Rudiments of μ-Calculus.* North-Holland, 2001.

[3] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis. On the desirability of acyclic database schemes. *Journal of the ACM*, 30:479–513, 1983.

[4] C. Berge. *Graphs and Hypergraphs.* North-Holland, 1973.

[5] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge Univ. Press, 2001.

[6] G. Boolos and G. Sambin. An incomplete system of modal logic. *Journal of Philosophical Logic*, 14:351–358, 1985.

[7] R. Bull and K. Segerberg. Basic modal logic. In Gabbay and Guenthner [33], pages 1–88.

[8] R. Carnap. *Logical Foundations of Probability.* Univ. of Chicago Press, Chicago, 1950.

[9] A. Chagrov, F. Wolter, and M. Zakharyaschev. Advanced modal logic. In *Handbook of Philosophical Logic, vol 3*, pages 83–266. Kluwer Academic Publishers, second edition, 2001.
[10] A. Chagrov and M. Zakharyaschev. *Modal Logic*. Oxford University Press, Oxford, 1997.
[11] L. Chagrova. An undecidable problem in correspondence theory. *Journal of Symbolic Logic*, 56:1261–1272, 1991.
[12] C. Chang and H. Keisler. *Model Theory*. North-Holland, Amsterdam, 1973.
[13] W. Conradie. Definability and changing perspectives. The Beth property for three extensions of modal logic. Technical Report MoL-2002-03, ILLC Publications, Master of Logic Series, 2002.
[14] W. Conradie, V. Goranko, and D. Vakarelov. Elementary canonical formulae: a survey on syntactic, algorithmic, and model-theoretic aspects. In *M. Reynolds et al (eds.), Advances in Modal Logic, volume 5, Kings College Publ., London*, pages 17–51, 2005.
[15] G. D'Agostino. *Modal Logic and Non-Well-Founded Set Theory: Translation, Bisimulation, Interpolation*. PhD thesis, University of Amsterdam, 1998. ILLC Dissertation Series DS-1998-04.
[16] G. D'Agostino and M. Hollenberg. Logical questions concerning the μ-calculus: interpolation, Lyndon and Łoś-Tarski. *Journal of Symbolic Logic*, 65:310–332, 2000.
[17] A. Dawar and M. Otto. Modal characterisation theorems over special classes of frames. In *Proc. of LICS'2005, IEEE Press*, pages 21–30, 2005.
[18] M. de Rijke. *Extending Modal Logic*. PhD thesis, ILLC, University of Amsterdam, 1993.
[19] M. de Rijke. A Lindström theorem for modal logic. In Ponse et al. [109], pages 217–230.
[20] M. de Rijke. Modal model theory. Technical Report CS-R9517, CWI, Amsterdam, 1995.
[21] M. de Rijke. A note on graded modal logic. *Studia Logica*, 64(2):271–283, 2000.
[22] M. de Rijke and H. Sturm. Global definability in basic modal logic. In H. Wansing, editor, *Essays on Non-Classical Logic*, pages 111–135. World Scientific Publishers, 2001.
[23] K. Doets. *Basic Model Theory*. CSLI Publications, Stanford, California, 1996.
[24] K. Doets and J. van Benthem. Higher-order logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, vol. 1*, pages 275–329. Reidel, Dordrecht, 1983.
[25] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Perspectives in Mathematical Logic. Springer, Berlin, 1995.
[26] H.-D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic*. Springer, Berlin, 1984.
[27] R. Fagin. Probabilities on finite models. *Journal of Symbolic Logic*, 41:50–58, 1976.
[28] K. Fine. Some connections between elementary and modal logic. In Kanger [81], pages 15–31.
[29] M. Fischer and R. Ladner. Propsitional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18:194–211, 1979.
[30] J. Flum. On the (infinite) model theory of fixed-point logics. Preprint, Inst. für Math. Logik und Grundl. der Math., Albert-Ludwigs-Universität Freiburg, December 1995.
[31] D. Gabbay. Selective filtration in modal logic. *Theoria*, 30:323–330, 1970.
[32] D. Gabbay. *Investigations in Modal and Tense Logics with Applications to Problems in Philosophy and Linguistics*. Reidel, Dordrecht, 1976.
[33] D. Gabbay and F. Guenthner, editors. *Handbook of Philosophical Logic, vol. 2*. Reidel, Dordrecht, 1984.
[34] D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*. Oxford University Press, 1994.
[35] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-dimensional modal logics: theory and applications*. Studies in Logic, 148. Elsevier Science, 2003.
[36] D. Gabbay and L. Maksimova. *Interpolation and Definability. Modal and Intuitionistic Logic*. Oxford Logic Guides, 46. Oxford Univ. Press, 2005.
[37] H. Gaifman. Concerning measures in first-order calculi. *Israel J. of Math.*, 2:1–18, 1964.
[38] H. Gaifman. On local and nonlocal properties. In J. Stern, editor, *Logic Colloquium '81*, pages 105–135. North Holland, 1982.
[39] S. Ghilardi and M. Zawadowski. *Sheaves, Games, and Model Completions*. Trends in Logic, 14. Kluwer, 2002.
[40] Y. Glebskii, D. Kogan, M. Liogonki, and V. Talanov. Range and degree of realizability of formulas in the restricted predicate calculus. *Cybernetics*, 5:142–154, 1969.
[41] G. Goguadze, C. Piazza, and Y. Venema. Simulating polyadic modal logics by monadic ones. *Journal of Symbolic Logic*, 68:419–462, 2003.
[42] R. Goldblatt. First-order definability in modal logic. *Journal of Symbolic Logic*, 40:35–40, 1975.
[43] R. Goldblatt. Metamathematics of modal logic I. *Reports on Mathematical Logic*, 6:41–78, 1976.
[44] R. Goldblatt. Metamathematics of modal logic II. *Reports on Mathematical Logic*, 7:21–52, 1976.
[45] R. Goldblatt. Varieties of complex algebras. *Annals of Pure and Applied Logic*, 38:173–241, 1989.
[46] R. Goldblatt. *Logics of Time and Computation*. Number 7 in CSLI Lecture Notes. CSLI, Stanford, 2nd edition, 1992.
[47] R. Goldblatt. *Mathematics of Modality*, volume 43 of *CSLI Lecture Notes*. Center for the Study of Language and Information, Stanford University, 1993.

[48] R. Goldblatt. Saturation and the Hennessy-Milner property. In Ponse et al. [109], pages 107–129.
[49] R. Goldblatt. Mathematical modal logic: A view of its evolution. *Journal of Applied Logic*, 1:309–392, 2003.
[50] R. Goldblatt, I. Hodkinson, and Y. Venema. Erdös graphs resolve Fine's canonicity problem. *Bulletin of Symbolic Logic*, 10:186–208, 2004.
[51] R. Goldblatt and S. Thomason. Axiomatic classes in propositional modal logic. In J. Crossley, editor, *Algebra and Logic*, pages 163–173. Springer, 1974.
[52] V. Goranko. Axiomatizations with context rules of inference in modal logic. *Studia Logica*, 61:179–197, 1998.
[53] V. Goranko and B. Kapron. The modal logic of the countable random frame. *Archive for Mathematical Logic*, 42:221–243, 2003.
[54] V. Goranko and S. Passy. Using the universal modality: Gains and questions. *Journal of Logic and Computation*, 2:5–30, 1992.
[55] V. Goranko and D. Vakarelov. Elementary canonical formulae. Extending Sahlqvist's theorem. *Annals of Pure and Applied Logic*, 141:180–217, 2005.
[56] E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64:1719–1742, 1999.
[57] E. Grädel. Why are modal logics so robustly decidable? *Bulletin of the EATCS*, 68:90–103, 1999.
[58] E. Grädel, C. Hirsch, and M. Otto. Back and forth between guarded and modal logics. *ACM Transactions on Computational Logics*, 3(3):418 – 463, 2002.
[59] E. Grädel, P. Kolaitis, and M. Vardi. On the Decision Problem for Two-Variable First-Order Logic. *Bulletin of Symbolic Logic*, 3:53–69, 1997.
[60] E. Grädel and M. Otto. On Logics with Two Variables. *Theoretical Computer Science*, 224:73–113, 1999.
[61] E. Grädel, M. Otto, and E. Rosen. Undecidability Results on Two-Variable Logics. *Archive for Mathematical Logic*, 38:213–354, 1999.
[62] E. Grädel and I. Walukiewicz. Guarded Fixed Point Logic. In *Proceedings of 14th IEEE Symposium on Logic in Computer Science LICS '99, Trento*, pages 45–54, 1999.
[63] E. Grandjean. Complexity of the first–order theory of almost all structures. *Information and Computation*, 52:180–204, 1983.
[64] Y. Gurevich. Zero-one laws: The logic in computer science column. *Bulletin of the European Association for Theoretical Computer Science*, 46:90–106, 1992.
[65] T. Hafer and W. Thomas. Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. In *Proceedings of ICALP 1987*, pages 269–279, 1987.
[66] J. Halpern and B. Kapron. Zero-one laws for modal logic. *Annals of Pure and Applied Logic*, 69:157–193, 1994.
[67] B. Herwig and D. Lascar. Extending partial isomorphisms and the profinite topology on free groups. *Transactions of the AMS*, 352:1985–2021, 2000.
[68] W. Hodges. *Model Theory*. Cambridge University Press, Cambridge, 1993.
[69] I. Hodkinson. Loosely guarded fragment of first-order logic has the finite model property. *Studia Logica*, 70:205–240, 2002.
[70] I. Hodkinson and M. Otto. Finite conformal hypergraph covers and Gaifman cliques in finite structures. *The Bulletin of Symbolic Logic*, 9:387–405, 2003.
[71] M. Hollenberg. Hennessy-Milner classes and process algebra. In Ponse et al. [109], pages 187–216.
[72] M. Hollenberg. Safety for bisimulation in general modal logic. In *Proceedings 10th Amsterdam Colloquium*, pages 359–376, 1996.
[73] E. Hoogland. *Definability and Interpolation: Model-theoretic investigations*. PhD thesis, University of Amsterdam, 2001. ILLC Dissertation Series DS-2001-05.
[74] G. Hughes. Every world can see a reflexive world. *Studia Logica*, 49:175–181, 1990.
[75] G. Hughes and M. Cresswell. *A New Introduction to Modal Logic*. Routledge, London, 1996.
[76] N. Immerman. Languages that capture complexity classes. *SIAM Journal of Computing*, 16:760–778, 1987.
[77] D. Janin and I. Walukiewicz. On the expressive completeness of the propositional μ-calculus w.r.t. monadic second-order logic. In *Proceedings CONCUR '96*, pages 263–277, 1996.
[78] B. Jónsson and A. Tarski. Boolean algebras with operators, Part I. *American Journal of Mathematics*, 73:891–939, 1952.
[79] J. Jónsson and A. Tarski. Boolean algebras with operators, Part II. *American Journal of Mathematics*, 74:127–162, 1952.
[80] H. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, 1968.
[81] S. Kanger, editor. *Proceedings of the Third Scandinavian Logic Symposium. Uppsala 1973.*, Amsterdam, 1975. North-Holland.
[82] C. Karp. *Languages with Expressions of Infinite Length*. North-Holland, Amsterdam, 1964.
[83] H. Keisler. *Model Theory for Infinitary Logic*. North-Holland, Amsterdam, 1971.

[84] M. Kracht. *Tools and Techniques in Modal Logic*, volume 142 of *Studies in Logic*. Elsevier, Amsterdam, 1999.

[85] M. Kracht and F. Wolter. Simulation and transfer results in modal logic: a survey. *Studia Logica*, 59:149–177, 1997.

[86] S. Kripke. A completeness theorem in modal logic. *Journal of Symbolic Logic*, 24:1–14, 1959.

[87] S. Kripke. Semantical analysis of modal logic I. Normal modal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.

[88] N. Kurtonina and M. de Rijke. Expressiveness of concept expressions in first-order description logics. *Artificial Intelligence*, 107(2):303–333, 1999.

[89] A. Lachlan. A note on Thomason's refined structures for tense logics. *Theoria*, 40:117–120, 1970.

[90] J.-M. Le Bars. Zero-one law fails for frame satisfiability in propositional modal logic. In *Proceedings of LICS'2002*, pages 225–234. IEEE Computer Society Press, 2002.

[91] E. Lemmon and D. Scott. *An Introduction to Modal Logic*. Blackwell, Oxford, 1977.

[92] D. Lewis. Intensional logics without iterative axioms. *Journal of Philisophical Logic*, 3:457–466, 1974.

[93] L. Libkin. *Elements of Finite Model Theory*. Springer, 2004.

[94] A. Lisitsa and V. Sazonov. Linear ordering on graphs, anti-founded sets and polynomial time computability. *Theoretical Computer Science*, 224:173–213, 1999.

[95] E. Lopez-Escobar. On defining well-orderings. *Fundamenta Mathematicae*, 59:299–300, 1966.

[96] C. Lutz, U. Sattler, and F. Wolter. Modal logic and the two-variable fragment. In *Proceedings of 15th International Workshop on Computer Science Logic*, volume 2142 of *LNCS*, pages 247–261, 2001.

[97] D. Makinson. A generalization of the concept of relational model. *Theoria*, pages 331–335, 1970.

[98] L. Maksimova. Definability and interpolation in classical modal logics. *Contemporary Mathematics*, 131(3):583–599, 1992.

[99] M. Marx. Complexity of modal logics of relations. Technical Report ML–97–02, Institute for Logic, Language and Computation, University of Amsterdam, May 1997.

[100] M. Marx. Xpath and modal logics of finite DAG's. In *Automated Reasoning with Analytic Tableaux and Related Methods, LNCS 2796*, pages 150–164, 2003.

[101] F. Moller and A. Rabinovich. On the expressive power of CTL*. In *Proceedings of the 14th IEEE Symposium on Logic in Computer Science*, pages 360–369, 1999.

[102] M. Mortimer. On languages with two variables. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 21:135–140, 1975.

[103] A. Nonnengart and A. Szalas. A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. In E. Orlowska, editor, *Logic at Work, Essays dedicated to the memory of Helena Rasiowa*, pages 89–108. Springer Physica-Verlag, 1998.

[104] M. Otto. Bisimulation-invariant Ptime and higher-dimensional mu-calculus. *Theoretical Computer Science*, 224:237–265, 1999.

[105] M. Otto. Elementary proof of the van Benthem–Rosen characterisation theorem. Technical Report 2342, Fachbereich Mathematik, Technische Universität Darmstadt, 2004.

[106] M. Otto. Modal and guarded characterisation theorems over finite transition systems. *Annals of Pure and Applied Logic*, 130:173–205, 2004.

[107] M. Otto. Bisimulation invariance and finite models. to appear in: Lecture Notes in Logic, Logic Colloquium 02, 2005.

[108] B. Poizat. *A Course in Model Theory*. Universitext. Springer, 2000.

[109] A. Ponse, M. de Rijke, and Y. Venema, editors. *Modal Logic and Process Algebra. A Bisimulation Perspective.*, volume 53 of *CSLI Lecture Notes*, Stanford University, 1995. CSLI Publications.

[110] S. Radev. Infinitary propositional normal modal logic. *Studia Logica*, 46:291–309, 1987.

[111] M. Reynolds. An axiomatization of full computation tree logic. *Journal of Symbolic Logic*, 66:1011–1057, 2001.

[112] E. Rosen. Modal logic over finite structures. *Journal of Logic, Language and Information*, 6:427–439, 1997.

[113] H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logic. In Kanger [81], pages 110–143.

[114] G. Sambin and V. Vaccaro. Topology and duality in modal logic. *Annals of Pure and Applied Logic*, 37:249–296, 1988.

[115] G. Sambin and V. Vaccaro. A topological proof of Sahlqvist's theorem. *Journal of Symbolic Logic*, 54:992–999, 1989.

[116] D. Scott. A decision method for validity of sentences in two variables. *Journal of Symbolic Logic*, 27:477, 1962.

[117] K. Segerberg. *An Essay in Classical Modal Logic*. Filosofiska Studier 13. University of Uppsala, 1971.

[118] R. Streett and E. Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Information and Computation*, 81:249–264, 1989.

[119] H. Sturm. *Modale Fragmente von $\mathcal{L}_{\omega\omega}$ und $\mathcal{L}_{\omega_1\omega}$*. PhD thesis, CIS, University of Munich, 1997.

[120] H. Sturm. Interpolation and preservation in ML_{ω_1}. *Notre Dame Journal of Formal Logic*, 39:190–211, 1998.

[121] B. ten Cate. Expressivity of second-order propositional modal logic. *Journal of Philosophical Logic*, 2005. to appear.

[122] B. ten Cate. *Model theory for extended modal languages*. PhD thesis, University of Amsterdam, 2005. ILLC Dissertation Series DS-2005-01.

[123] S. Thomason. An incompleteness theorem in modal logic. *Theoria*, 40:150–158, 1974.

[124] S. Thomason. Reduction of second-order logic to modal logic. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 21:107–114, 1975.

[125] J. van Benthem. *Modal Correspondence Theory*. PhD thesis, Mathematisch Instituut & Instituut voor Grondslagenonderzoek, University of Amsterdam, 1976.

[126] J. van Benthem. Canonical modal logics and ultrafilter extensions. *Journal of Symbolic Logic*, 44:1–8, 1980.

[127] J. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, Naples, 1983.

[128] J. van Benthem. Correspondence theory. In Gabbay and Guenthner [33], pages 167–247.

[129] J. van Benthem. Notes on modal definability. *Notre Dame Journal of Formal Logic*, 39:20–39, 1989.

[130] J. van Benthem. *Exploring Logical Dynamics*. Studies in Logic, Language and Information. CSLI Publications, Stanford, 1996.

[131] J. van Benthem. Modality, bisimulation and interpolation in infinitary logic. Technical Report ML-97-06, Institute for Logic, Language and Computation, University of Amsterdam, 1997.

[132] J. van Benthem. Minimal predicates, fixed points, and definability. *Journal of Symbolic Logic*, 70:696–712, 2005.

[133] J. van Benthem. A new Lindström theorem for modal logic. ILLC Publication Series Report PP-2006-06, 2006.

[134] M. Vardi. The complexity of relational query languages. In *Proceedings of 14th ACM Symposium on Theory of Computing*, pages 137–146, 1982.

[135] M. Vardi. Why is modal logic so robustly decidable? In N. Immerman and P. Kolaitis, editor, *Descriptive Complexity and Finite Models*, volume 31 of *DIMACS Series in Discrete Mathematics*, pages 149–184. AMS, 1997.

[136] M. Vardi. Reasoning about the past with two-way automata. In *Proceedings of 25th Symposium on Automata, Language and Programming*, volume 1443 of *LNCS*, pages 628–641, 1998.

[137] Y. Venema. Derivation rules as anti-axioms in modal logic. *Journal of Symbolic Logic*, 58:1003–1034, 1993.

[138] Y. Venema. Modal definability, purely modal, 1999. in: J Gerbrandy et al. (editors), Essays Dedicated to Johan van Benthem on the Occasion of his 50th Birthday.

[139] A. Visser. Bisimulations, model descriptions and propositional quantifiers. Technical Report 161, Department of Philosophy, Utrecht University, 1996.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

6

ALGEBRAS AND COALGEBRAS

Yde Venema

1 Introduction . . . 332
2 Basics of modal logic . . . 333
3 Modal logic in algebraic form . . . 337
4 Varieties of expanded Boolean algebras . . . 342
5 Frames and algebras . . . 347
  5.1 Introduction . . . 347
  5.2 Complex duality . . . 348
  5.3 Ultrafilter frames . . . 351
  5.4 Topological duality . . . 354
  5.5 Simplicity and Subdirect irreducibility . . . 356
  5.6 Class operations . . . 358
6 Logics and varieties . . . 361
  6.1 Completeness . . . 362
  6.2 Canonicity . . . 365
  6.3 Interpolation . . . 369
7 Case study: canonical equations . . . 371
  7.1 Introduction . . . 371
  7.2 Canonical extensions of Boolean algebras . . . 373
  7.3 Extending maps to the canonical extension . . . 375
  7.4 Composite maps . . . 377
  7.5 Canonical equations . . . 380
  7.6 Further remarks . . . 383
8 Special algebraic topics . . . 384
  8.1 Tense logic . . . 384
  8.2 Global modality & discriminator varieties . . . 386
9 Coalgebras: an introduction . . . 389
10 Final coalgebras . . . 395
11 Bisimulation & behavioral equivalence . . . 398
12 Covarieties . . . 403
13 Modal logic and coalgebras . . . 406
14 Co-Birkhoff theorems and cofree coalgebras . . . 412
15 Duality of algebra and coalgebra . . . 417
A Basics of universal algebra and category theory . . . 418

1 INTRODUCTION

Modal logic is not an isolated field. When studied from a mathematical perspective, it has evident connections with many other areas in logic, mathematics and theoretical computer science. Other chapters of this handbook point out some of the links between modal logic and areas like (finite) model theory or automata theory. Here we will outline the *algebraic* and *coalgebraic* environments of the theory of modal logic.

First we approach modal logic with the methodology of *algebraic logic*, a discipline which aims at studying all kinds of logics using tools and techniques from universal algebra — in fact, much of the theory of universal algebra was developed in tandem with that of algebraic logic. The idea is to associate, with any logic L, a class $\mathsf{Alg}(L)$ of algebras, in such a way that (natural) logical properties of L correspond to (natural) algebraic properties of $\mathsf{Alg}(L)$. Carrying out this program for modal logic, we find that normal modal logics have algebraic counterparts in varieties of *Boolean algebras with operators* (BAOs). In the simplest case of monomodal logics, the algebras that we are dealing with are simply *modal algebras*, that is, expansions of Boolean algebras with a single, unary operation that preserves finite joins (disjunctions). One advantage of the algebraic semantics over the relational one is that it allows a general *completeness* result, but the algebraic approach may also serve to prove many significant results concerning properties of modal logics such as completeness, canonicity, and interpolation. As we will see, a crucial observation in the algebraic theory of modal logic is that standard algebraic constructions correspond to well-known operations on Kripke frames. These correspondences can be made precise in the form of categorical *dualities*, which may serve to explain much of the interaction between modal logic and universal algebra. Our discussion of the algebraic approach towards modal logics takes up the sections 3 to 8.

The *coalgebraic* perspective on modal logic is much more recent (see section 9 for references). Coalgebras are simple but fundamental mathematical structures that capture the essence of dynamic or evolving systems. The theory of universal coalgebra seeks to provide a general framework for the study of notions related to (possibly infinite) behavior such as invariance, and observational indistinguishability. When it comes to modal logic, an important difference with the algebraic perspective is that coalgebras *generalize* rather than *dualize* the model theory of modal logic. Many familiar notions and constructions, such as bisimulations and bounded morphisms, have analogues in other fields, and find their natural place at the level of coalgebra. Perhaps even more important is the realization that one may generalize the concept of modal logic from Kripke frames to arbitrary coalgebras. In fact, the link between (these generalizations of) modal logic and coalgebra is so tight, that one may even claim that modal logic is the natural logic for coalgebras — just like equational logic is that for algebra. The second and last part of this chapter, starting from section 9, is devoted to coalgebra.

What is the point of taking such an abstract perspective on modal logic, be it algebraic or coalgebraic? Obviously, making the above kind of mathematical generalizations, one should not aim at solving all concrete problems for specific modal logics. Rather, the approach may serve to isolate those aspects of a problem that are easy in the sense of being solvable by general means; it thus enables us to focus on the remaining aspects that are specific to the problem at hand. To give an example, it is certainly not the case that all modal formulas are canonical, but Sahlqvist's theorem considerably simplifies completeness proofs by taking care of the canonical part of the axiomatization. A second

benefit of embedding modal logic in its mathematical context is that it may lead to a better understanding of notions from modal logic. Taking an example from coalgebra, the notion of a bounded morphism between Kripke models (or frames), becomes much more natural once we understand that it coincides with the natural coalgebraic notion of a homomorphism.

Our main aim with this chapter is to give the reader an impression of both the algebraic and the coalgebraic perspective on modal logic. Our focus will be on concepts and ideas, but we will also mention important techniques and landmark results; proofs, or rather proof sketches, are given as much as possible. Despite its over-average length, a text of this size cannot come close to being comprehensive; our main selection criterion has been to focus on *generality* of methods and results. Unfortunately, even some important topics have fallen prey to this, most particularly, the *algebras of relations*, even though they played and continue to play a crucial role in the history of algebraic logic. Fortunately, these kinds of BAOs are well documented elsewhere, see for instance HENKIN, MONK & TARSKI [57] for cylindric algebras, or HIRSCH & HODKINSON [58] for relation algebras. A second topic receiving only fragmented attention is *historical context*. While we do attribute results as much as possible, readers with an interest in the (fascinating!) history of modal logic, will not find much to suit their taste here. Rather, they should consult GOLDBLATT [44], or perhaps the historical notes of BLACKBURN, DE RIJKE & VENEMA [13]. Finally, a warning: in this chapter we assume familiarity with basic notions from category theory (such as functors, duality), universal algebra (such as congruences, free algebras), and more specifically, Boolean algebras. Readers encountering unfamiliar concepts in this chapter are advised to consult some text book in universal algebra or category theory. For convenience, in an appendix we have summed up all the material that we consider to be background knowledge.

2 BASICS OF MODAL LOGIC

In this section we briefly review the basic definitions of modal logic. Starting with syntax, we take a fairly general approach towards modal languages and allow modal connectives of arbitrary finite rank. A *modal similarity type* is a set τ of modal connectives, together with an arity function $ar : \tau \to \omega$ assigning to each symbol $\nabla \in \tau$ a *rank* or *arity* $ar(\nabla)$. Given a modal similarity type τ and a set X of variables we inductively define the set $Fma_\tau(X)$ of *modal τ-formulas in X* by the following rule:

$$\varphi ::= x \in X \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \nabla(\varphi_1, \ldots, \varphi_n)$$

with $\nabla \in \tau$ and $n = ar(\nabla)$. We will use standard abbreviations such as $\to$ and $\leftrightarrow$; we also define the *dual* operator Δ of $\nabla \in \tau$ as $\Delta(\varphi_1, \ldots, \varphi_n) := \neg\nabla(\neg\varphi_1, \ldots, \neg\varphi_n)$. Unary modalities are usually called *diamonds*, and their duals, *boxes*; to denote these modalities we reserve (possibly indexed) symbols of the shape $\Diamond$ and $\Box$, respectively.

Throughout this chapter we will work with an arbitrary but fixed modal similarity type τ. Often, we will provide proofs only for the *basic modal similarity type* which consists of a single diamond that will always simply be denoted as $\Diamond$ (its dual as $\Box$). Unless explicitly stated otherwise, we are always dealing with a fixed, countably infinite set X of variables; in order not to clutter up notation we will suppress explicit references to X as much as possible.

It will be convenient to have names and notation for some special formulas that behave just like ordinary diamond formulas of the form $\Diamond v$. Fix a special dummy variable v. In the basic modal language, we may define a *compound diamond* as any disjunction of formulas of the form $\Diamond^n v$ (here $\Diamond^0\varphi := \varphi$ and $\Diamond^{n+1}\varphi := \Diamond\Diamond^n\varphi$). In a language with diamonds only, the compound diamonds may be defined as follows:

$$\blacklozenge ::= v \mid \Diamond_i\blacklozenge \mid \blacklozenge_1 \vee \blacklozenge_2.$$

An example is $\Diamond_1\Diamond_0(\Diamond_1 v \vee v) \vee \Diamond_1\Diamond_1 v$. We will write $\blacklozenge\varphi$ for the formula in which every occurrence of v is substituted by φ (note that v is the *unique* variable occurring in a compound diamond). Induced and compound *boxes* are defined in the obvious, dual, way.

The general case, which readers may safely choose to skip, is a bit more involved. For any modality ∇ of arity $n > 1$, and any $1 \leq i \leq n$, the formula $\Diamond_{(\nabla,i)}v := \nabla(\top,\ldots,\top,v,\top,\ldots,\top)$ (i.e., all arguments are $\top$ except for the i-th one which is v) is called the i-th *induced diamond* of ∇. The collection $CD(\tau)$ of *compound diamonds* of τ is defined via:

$$\blacklozenge ::= v \mid \Diamond_i\blacklozenge \mid \Diamond_{(\nabla,i)}\blacklozenge \mid \blacklozenge_1 \vee \blacklozenge_2.$$

Modal logic can be approached from a semantic or from a purely syntactic/axiomatic angle. In this chapter we follow both approaches, starting with the semantic one.

DEFINITION 1. A *τ-frame* is a structure $\mathbb{S} = \langle S, R\rangle$ where S is a non-empty set of objects called *states*, *points*, or *worlds*, and R is an interpretation assigning an $n+1$-ary relation R_∇ on S to every n-ary modal connective ∇. A *valuation* on $\mathbb{S}$ is a map $V : X \to \mathcal{P}(S)$ assigning a subset of S to each variable x. A *τ-model* is a structure $\mathbb{M} = \langle S, R, V\rangle$ such that $\langle S, R\rangle$ is a τ-frame, on which V is a valuation; the frame $\langle S, R\rangle$ is called the *underlying frame* of $\mathbb{M}$.

The notion of *truth* is defined by formula induction. The set of points where φ is true will always be denoted as $[\![\varphi]\!]$.

DEFINITION 2. Given a τ-model $\mathbb{M}$, we define by induction when a formula φ is *true* at a state s of $\mathbb{M}$, notation: $\mathbb{M}, s \Vdash \varphi$:

$$\begin{array}{rll}
\mathbb{M}, s \Vdash x & \text{if} & s \in V(x), \\
\mathbb{M}, s \Vdash \top & & \text{always}, \\
\mathbb{M}, s \Vdash \bot & & \text{never}, \\
\mathbb{M}, s \Vdash \neg\varphi & \text{if} & \mathbb{M}, s \not\Vdash \varphi, \\
\mathbb{M}, s \Vdash \varphi \wedge \psi & \text{if} & \mathbb{M}, s \Vdash \varphi \text{ and } \mathbb{M}, s \Vdash \psi, \\
\mathbb{M}, s \Vdash \varphi \vee \psi & \text{if} & \mathbb{M}, s \Vdash \varphi \text{ or } \mathbb{M}, s \Vdash \psi, \\
\mathbb{M}, s \Vdash \nabla(\varphi_1,\ldots,\varphi_n) & \text{if} & R_\nabla s s_1 \ldots s_n \text{ for some } s_1,\ldots,s_n \text{ such that } \mathbb{M}, s_i \Vdash \varphi_i \text{ for} \\
& & \text{all } i \leq n.
\end{array}$$

We write $\mathbb{M} \Vdash \varphi$ if φ is true *throughout* $\mathbb{M}$, that is, true at every state of $\mathbb{M}$.

DEFINITION 3. Given a τ-frame $\mathbb{S}$, we say that a modal formula φ is *valid* in $\mathbb{S}$, notation: $\mathbb{S} \Vdash \varphi$ if φ is true throughout any model based on $\mathbb{S}$. Similarly standard definitions apply to sets of formulas and classes of frames.

Using the notation $Q[s] := \{t \mid Qst\}$ for any binary relation Q, we define the relation $R_{\mathbb{S}}$ such that $R_{\mathbb{S}}[s]$ consists of those points that can be reached from s in one step using any of the accessibility relations, and R^{ω} as the reflexive and transitive closure of $R_{\mathbb{S}}$.

We may extend the interpretation R of a τ-frame $\mathbb{S} = \langle S, R\rangle$ to the compound diamonds by putting

$$\begin{array}{rcl} R_{v} & := & Id\ (= \{(s,s) \mid s \in S\}) \\ R_{\diamond_{(\nabla,i)}\blacklozenge} & := & \{(s, s_i) \mid R_{\nabla} s s_1 \cdots s_n \text{ for some } s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n \in S\} \circ R_{\blacklozenge}, \\ R_{\diamond\blacklozenge} & := & R_{\diamond} \circ R_{\blacklozenge}, \\ R_{\blacklozenge_1 \vee \blacklozenge_2} & := & R_{\blacklozenge_1} \cup R_{\blacklozenge_2}. \end{array}$$

It is then straightforward to verify, in any frame $\mathbb{S}$, that $R^{\omega} = \bigcup_{\blacklozenge \in CD(\tau)} R_{\blacklozenge}$, and that for any valuation V it holds that

$$\mathbb{S}, V, s \Vdash \blacklozenge\varphi \text{ iff } \mathbb{S}, V, t \Vdash \varphi \text{ for some } t \text{ with } R_{\blacklozenge}st.$$

That is, compound diamonds indeed behave like diamonds.

Frames and models do not exist in isolation. Given two τ-frames $\mathbb{S}$ and $\mathbb{S}'$, a map $\theta : S \to S'$ is called a *bounded morphism* from $\mathbb{S}$ to $\mathbb{S}'$, notation: $\theta : \mathbb{S} \to \mathbb{S}'$, if θ satisfies the following conditions for all $\nabla \in \tau$:

(forth) $R_{\nabla} s s_1 \ldots s_n$ only if $R'_{\nabla}\theta(s)\theta(s_1)\ldots\theta(s_n)$, and

(back) $R'_{\nabla}\theta(s)s'_1 \ldots s'_n$ only if there are $s_1, \ldots, s_n$ such that $R_{\nabla} s s_1 \ldots s_n$ and $\theta(s_i) = s'_i$ for each i.

We let Fr_{τ} denote the category with τ-frames as objects and bounded morphisms as arrows.

If such a bounded morphism θ is surjective, we call $\mathbb{S}'$ a *bounded morphic image* of $\mathbb{S}$, notation: $\mathbb{S} \twoheadrightarrow \mathbb{S}'$; if θ is injective we write $\mathbb{S} \rightarrowtail \mathbb{S}'$ and call the subframe of $\mathbb{S}'$ based on the image $\theta[S]$ a *generated subframe* of $\mathbb{S}'$. We leave it for the reader to verify that the structure $\langle T, R{\restriction}T\rangle$ (where $R{\restriction}T$ maps each $\nabla \in \tau$ to the *restriction* of R_{∇} to T) is a generated subframe of $\mathbb{S}$ if and only if T is a *hereditary* subset of S, that is, if $t \in T$ then $R_{\nabla} t t_1 \ldots t_n$ implies that all the t_i belong to T. Given a point r in $\mathbb{S}$, we denote with $\mathbb{S}_r$ the least generated subframe containing r; the domain of this subframe is thus the set $R^{\omega}[r]$. If $\mathbb{S} = \mathbb{S}_r$ we call r a *root* of $\mathbb{S}$, and say that $\mathbb{S}$ is *rooted*. Finally, given a *family* $\{\mathbb{S}_i \mid i \in I\}$ of τ-frames, we define its *disjoint union* $\sum_{i\in I} \mathbb{S}_i$ as the structure $\langle \sum_{i\in I} S_i, \{R_{\nabla} \mid \nabla \in \tau\}\rangle$, where the domain $\sum_{i\in I} S_i = \bigcup_{i\in I}\{i\} \times S_i$ is the disjoint union of the domains S_i, and the relation R_{∇} is given by $R_{\nabla}(i,s)(i_1,s_1)\ldots(i_n,s_n) \;:\Longleftrightarrow\; i = i_1 = \ldots = i_n$ and $(R_i)_{\nabla} s s_1 \ldots s_n$.

REMARK 4. More general than Kripke frames are the neighborhood frames, which we now review very briefly, and for the basic modal similarity type only. The reader can find more details on these structures in Chapter 1 of this volume. A *neighborhood frame* is a structure $\mathbb{S} = \langle S, \sigma\rangle$ with $\sigma : S \to \mathcal{PP}(S)$; such a structure is called *monotone* if $\sigma(s)$ is upwards closed for all $s \in S$, that is, $X \in \sigma(s)$ and $X \subseteq Y$ imply $Y \in \sigma(s)$. Elements of $\sigma(s)$ are called *neighborhoods* of s, and the semantics of the modality ∇ (we will not use $\diamond$ and $\Box$ in this context) in a *neighborhood model* $\mathbb{M} = \langle S, \sigma, V\rangle$ with $V : X \to \mathcal{P}(S)$ a valuation is given by

$$\mathbb{M}, s \Vdash \nabla\varphi \text{ if } [\![\varphi]\!] \in \sigma(s), \tag{1}$$

that is, $\nabla\varphi$ holds at s iff s has a neighborhood of φ-points. Both the box and the diamond interpretation in Kripke models follow the pattern of (1): take $\sigma_\Diamond(s) = \{A \subseteq S \mid A \cap R[s] \neq \varnothing\}$, and $\sigma_\Box(s) = \{A \subseteq S \mid R[s] \subseteq A\}$, respectively.

A map $f : S \to S'$ is a *neighborhood morphism* between two neighborhood frames $\langle S, \sigma\rangle$ and $\langle S', \sigma'\rangle$ if for all $s \in S$ and all $X' \subseteq S'$ it holds that $X' \in \sigma'(fs)$ iff $f^{-1}[X'] \in \sigma(s)$.

Now we turn to the more syntactic approach towards modal logic. We identify logics with sets of theorems — the more general approach based on *consequence relations* will be discussed in Chapter 8 of this book. A *modal τ-logic* is then a set $L \subseteq Fma_\tau$ which (i) contains all classical propositional tautologies, and (ii) is closed under the derivation rules (MP) of *Modus Ponens* (if both φ and $\varphi \to \psi$ belong to L then so does ψ), and (US) of *uniform substitution* (if φ belongs to L then so do all of its substitution instances). If a formula φ belongs to a modal logic L then we say that φ is a *theorem* of L, notation: $\vdash_L \varphi$.

A modal logic is called *classical* if it is closed under the following rule: $\vdash_L \varphi_i \leftrightarrow \psi_i \Rightarrow \vdash_L \nabla(\varphi_1, \ldots, \varphi_n) \leftrightarrow \nabla(\psi_1, \ldots, \psi_n)$; *monotone* if it is closed under $\vdash_L \varphi_i \to \psi_i \Rightarrow \vdash_L \nabla(\varphi_1, \ldots, \varphi_n) \to \nabla(\psi_1, \ldots, \psi_n)$; and *normal* if it contains in addition, for each $\nabla \in \tau$, the axioms $\neg\nabla^i\bot$ and $\nabla(\overline{p}, q \vee q', \overline{r}) \to \nabla(\overline{p}, q, \overline{r}) \vee \nabla(\overline{p}, q', \overline{r})$ where $\overline{p}$ and $\overline{r}$ denote arbitrary sequences of propositional variables of combined length $ar(\nabla) - 1$. We leave it as an exercise for the reader to verify that this definition coincides with the standard one in the case of basic modal logic.

The *minimal* classical, monotone and normal modal logics for a similarity type τ are denoted as $\mathbf{C}_\tau$, $\mathbf{M}_\tau$ and $\mathbf{K}_\tau$, respectively. Here we use the convention that $\mathbf{C}$, $\mathbf{M}$ and $\mathbf{K}$ denote the minimal logics for the basic modal similarity type. It is easy to see that the collection of normal modal logics is closed under taking arbitrary intersections and therefore forms a complete lattice under the inclusion ordering. Hence, with any set Γ of τ-formulas we may associate the *least* normal modal τ-logic extending $\mathbf{K}$ and containing all formulas in Γ; this logic is denoted as $\mathbf{K}_\tau.\Gamma$. We say that this logic is *axiomatized* by Γ, since any theorem in $\mathbf{K}_\tau.\Gamma$ can be obtained as the result of a *derivation* from the axioms of the logic (including formulas in Γ) using its derivation rules. Similar definitions and notation apply to extensions of $\mathbf{C}_\tau$ and $\mathbf{M}_\tau$.

The validity relation $\Vdash$ between frames and formulas induces a Galois connection consisting of two maps, *Log* and Fr, defined as follows. Given a class C of frames, $Log(\mathsf{C})$, the *logic of* C, is the set of modal formulas that are valid in C. Conversely, given a set Γ of formulas, let $\mathsf{Fr}(\Gamma)$ denote the class of frames on which Γ is valid. (We call this a Galois connection because we always have $\mathsf{C} \subseteq \mathsf{Fr}(\Gamma)$ iff $\Gamma \subseteq Log(\mathsf{C})$.) The *stable* sets of formulas of this connection, that is, the sets Γ such that $\Gamma = Log(\mathsf{Fr}(\Gamma))$ are called *(Kripke) complete logics* — we leave it for the reader to verify that such sets are indeed normal modal logics. On the other side, the *stable* frame classes, that is, the ones that are closed under the composition $\mathsf{Fr} \circ Log$, are called *(modally) definable*. Not all modal logics are Kripke complete (see Chapter 7 of this volume) and not all frame classes are modally definable (see Chapter 1 of this volume).

3 MODAL LOGIC IN ALGEBRAIC FORM

As indicated in the introduction, it is the aim of algebraic logic to study logic by algebraic means. Nowadays, most people will associate modal logic primarily with relational structures, but, as with other branches of logic, the 19th century infancy of modern symbolic modal logic was completely algebraic, see MacColl [82]). Somehow during the 20th century however, the traditions of algebraic logic and of modal logic got separated, and for decades proceeded without any interaction whatsoever. In particular, while Jónsson & Tarski [70] introduced not only Boolean algebras with operators and their representation over relational structures, but also the rudiments of canonicity and correspondence theory, this seminal work did not mention modal logic, and it was completely overlooked by modal logicians for many years. This is not to say that algebras were to remain absent from the modal logic tradition — they were introduced by Lemmon [80]. But only in the 1970s, probably with the discovery of the fundamental incompleteness of the relational semantics by Thomason [102], did universal algebraic (and topological) methods regain importance — as examples we mention Blok [14], Esakia [23], Goldblatt [37, 38], and Rautenberg [91]. And it would even have to wait until the 1990s before the algebraic and modal traditions would be completely rejoined, with collaborations between modal and algebraic logicians (leading to, for instance, the introduction of the guarded fragment in Andréka, van Benthem & Németi [7]), with modal logicians investigating algebras of relations from a modal perspective (Marx & Venema [84]), or with algebraic logicians responding to the modal tradition (Jónsson [69]). It is from this perspective that the algebraic part of this chapter has been written.

Before we explain how to *algebraize* modal logic using the key structures of Boolean algebras with operators (BAOs), let us first briefly introduce the algebraic perspective on (propositional) logic itself. Think of proposition letters as atomic objects referring to entities called propositions, and of connectives as function symbols to be interpreted as operations on propositions. Then notice the complete analogy between the definitions of formulas and terms, respectively, and already we have worked our way towards one of the key ideas underlying the algebraic approach towards (propositional) logic: *propositional formulas can be seen as algebraic terms denoting propositions.*

DEFINITION 5. Given a modal similarity type τ, we define its corresponding algebraic similarity type $Bool_\tau$ simply as the union of τ with the Boolean similarity type $Bool = \{\top, \bot, \neg, \wedge, \vee\}$.

We will use $\approx$ as the equality symbol of this algebraic language; as abbreviations we use $\not\approx$ and $\preceq$ in their standard meaning. Since the standard Boolean symbols are function symbols in this algebraic language, we will not use them to denote Boolean combination of equations. For that purpose we let the symbols & and $\Rightarrow$ denote conjunction and implication, respectively.

The set $Fma_\tau(X)$ of formulas over a set of variables X can then be identified with the set $Ter_{Bool_\tau}(X)$ of algebraic $Bool_\tau$-terms over X. More importantly, we may impose *algebraic structure* on formulas.

DEFINITION 6. The τ-*formula algebra* is the structure $\mathbb{F}\mathsf{ma}_\tau := \langle Fma_\tau, \{\heartsuit^{\mathbb{F}\mathsf{ma}_\tau} \mid \heartsuit \in Bool_\tau\}\rangle$, where for each (Boolean or modal) connective $\heartsuit$, its interpretation

$$\heartsuit^{\mathbb{F}\mathsf{ma}_\tau} : \ (\varphi_1, \ldots, \varphi_n) \mapsto \heartsuit(\varphi_1, \ldots, \varphi_n)$$

defines a map of arity $n = ar(\heartsuit)$ on Fma_τ.

As a first advantage of this algebraic point of view, recall that substitutions are completely determined by their values on the variables. Putting this algebraically, for any function σ assigning formulas to variables, the substitution induced by σ is the unique extension $\widetilde{\sigma}$ of σ to an *endomorphism* on the formula algebra. More generally, it is easy to see that given an arbitrary algebra $\mathbb{A}$ of type $Bool_\tau$, any assignment mapping variables to elements of the carrier of $\mathbb{A}$ has a unique extension $\widetilde{\alpha}$ which is a *homomorphism* from $\mathbb{F}\mathsf{ma}_\tau$ to $\mathbb{A}$. That is, we have the following result.

PROPOSITION 7. $\mathbb{F}\mathsf{ma}_\tau$ *is the* ω*-generated absolutely free algebra of the similarity type* $Bool_\tau$.

Logical languages may now be interpreted in many different kinds of algebras; but of course, we are only interested in structures that can plausibly be viewed as algebras of propositions.

EXAMPLE 8. Consider the *truth value* algebra $\mathbb{2}$ of the Boolean similarity type. Its carrier is given as the set $2 = \{0, 1\}$ where 0 ('false') and 1 ('true') are the classical truth values, while its interpretation of the Boolean connectives/function symbols is given by the standard *truth tables*. Given a *valuation* $V : X \to 2$ of truth values to propositional variables, we can simply *compute* the truth value $\widetilde{V}(\varphi)$ of any propositional formula φ, using the unique homomorphism $\widetilde{V} : \mathbb{F}\mathsf{ma}_\tau \to \mathbb{2}$ extending the assignment V. That is, we see another manifestation of the absolute freeness of the formula algebra.

The algebras arising from the relational semantics of modal languages are the so-called complex algebras. (This terminology dates back to the times when subsets of groups were referred to as *complexes* of the group.)

DEFINITION 9. Given an $n+1$-ary relation R on a set S, define the n-ary map $\langle R\rangle$ on the power set of S by

$$\langle R\rangle(a_1, \ldots, a_n) := \{s \in S \mid Rss_1 \ldots s_n \text{ for some } s_1, \ldots, s_n \text{ with } s_i \in a_i \text{ for all } i\}.$$

The *complex algebra* $\mathbb{S}^+$ of a τ-frame $\mathbb{S}$ is obtained by expanding the power set algebra $\mathbb{P}(S)$ with operations $\langle R_\nabla\rangle$ for each modal connective ∇; that is,

$$\mathbb{S}^+ := \langle \mathcal{P}(S), S, \varnothing, \sim_S, \cap, \cup, \{\langle R_\nabla\rangle \mid \nabla \in \tau\}\rangle. \tag{2}$$

Given a frame class C, we let $\mathsf{Cm}(\mathsf{C})$ denote the class of complex algebras of frames in C; conversely, for a class K of algebras, $\mathsf{Str}(\mathsf{K})$ denotes the class of frames whose complex algebras belong to K.

REMARK 10. More generally, given a neighborhood frame $\mathbb{S} = \langle S, \sigma\rangle$, define the map $\sigma^+ : \mathcal{P}(S) \to \mathcal{P}(S)$ by $\sigma^+(A) := \{s \in S \mid A \in \sigma(s)\}$, and define $\mathbb{S}^+$ as the expansion of $\mathbb{P}(S)$ with the operation σ^+.

From the perspective of complex algebras, a *valuation* is nothing but an *assignment* of variables to elements of the complex algebra of $\mathbb{S}^+$. Furthermore, and much more importantly, given a valuation V on a frame $\mathbb{S}$, a straightforward induction proves that

$$\mathbb{S}, V, s \Vdash \varphi \text{ iff } s \in \widetilde{V}(\varphi), \tag{3}$$

where $\widetilde{V} : \mathbb{F}\mathsf{ma}_\tau \to \mathbb{S}^+$ is the unique homomorphism extending V. With the *meaning function* $[\![\cdot]\!]^{\mathbb{S},V}$ defined as the function that maps a formula φ to its *extension* $[\![\varphi]\!] := \{s \in S \mid \mathbb{S}, V, s \Vdash \varphi\}$, what (3) reveals is that, in a slogan, *meaning is a homomorphism*:

PROPOSITION 11. *Let V be some valuation on a τ-frame $\mathbb{S}$. Then the meaning function $[\![\cdot]\!]^{\mathbb{S},V}$ is the unique homomorphism $\widetilde{V} : \mathbb{F}\mathsf{ma}_\tau \to \mathbb{S}^+$ that extends V.*

As a corollary of this, let $\varphi^\approx$ denote the equation $\varphi \approx \top$, then we find that for any τ-frame $\mathbb{S}$, and any τ-formulas φ, ψ:

$$\mathbb{S} \Vdash \varphi \text{ iff } \mathbb{S}^+ \models \varphi^\approx \quad \text{and} \quad \mathbb{S} \Vdash \varphi \leftrightarrow \psi \text{ iff } \mathbb{S}^+ \models \varphi \approx \psi, \tag{4}$$

that is, the validity of a *formula* in the frame $\mathbb{S}$ corresponds to that of an *equation*[1] in the complex algebra of $\mathbb{S}$, and vice versa. We have arrived at one of the most fundamental notion of algebraic logic, namely, that of a class of algebras *algebraizing* a logic.

DEFINITION 12. Let L be a modal τ-logic, and K a class of $Bool_\tau$-algebras. We say that K *algebraizes* L, if we have

$$\vdash_L \varphi \quad \text{iff} \quad \mathsf{K} \models \varphi^\approx, \tag{5}$$

$$\mathsf{K} \models \varphi \approx \psi \quad \text{iff} \quad \vdash_L \varphi \leftrightarrow \psi, \tag{6}$$

for all formulas/terms φ and ψ.

The point of this definition is to alert the reader that algebraizations constitute stronger links between logics and classes of algebras than the mere existence of a completeness result, as would be expressed by (5) on its own. If the class K algebraizes the modal logic L, then it is not just the case that K contains all the information of L through the translation $(\cdot)^\approx$, but also, L encodes the full equational theory of K through the translation mapping an equation $\varphi \approx \psi$ to the formula $\varphi \leftrightarrow \psi$. Furthermore, the second translation is an *inverse* to the first one in the sense that if we translate the formula φ back and forth, the result $\varphi \leftrightarrow \top$ is L-equivalent to φ. Given the Boolean backbone of modal logics, this property holds vacuously, so there is no need to formulate this as an additional clause of the definition.

Also, observe that it immediately follows from the definition that if K algebraizes L, then so does the variety generated by K.

REMARK 13. The above definition is a specific instantiation of a much wider notion, which is due to Blok & Pigozzi [16]. The basic idea of a class of algebras *algebraizing* a logic always involves uniform translations from formulas to equations, and from equations to formulas, that are, modulo equivalence, inverse to each other. But the general case is of course not limited to modal logics, or to logics extending classical propositional logic; also, the translations may be from formulas to sets of equations, and from equations to sets of formulas.

The most important point is however that the natural habitat of the concept is that of *consequence relations* rather than of logics (in our sense of the word, that is, of logics as sets of sentences). In this more general setting, the requirement that the translations are each other's inverse, is expressed on the logical side by means of the consequence relation, and can equivalently be described on the algebraic side using (infinitary versions of) quasi-equations. For more details on modal consequence relations and the way to algebraize them, the reader is referred to Chapter 8 of this volume. For the general theory of algebraizing logics, see Czelakowski [21] or Font & Jansana [26].

[1] In the sequel, we will be sloppy about the distinction between a formula and its equational translation, writing for instance $\mathbb{A} \models \varphi$ if we mean $\mathbb{A} \models \varphi^\approx$.

In any case, it will be clear that we can already state our first algebraization result, the proof of which is immediate from (4):

THEOREM 14. *Let* C *be a class of* τ*-frames. Then* $\mathsf{Cm}(\mathsf{C})$ *algebraizes* $Log(\mathsf{C})$.

Turning to the algebraization of arbitrary modal logics, we now introduce the key players: *Boolean algebras with operators*, together with some related concepts.

DEFINITION 15. Given two Boolean algebras $\mathbb{B}$ and $\mathbb{B}'$, it is often convenient to call a function $f : B \to B'$ a *map from* $\mathbb{B}$ *to* $\mathbb{B}'$. Such a map is called *monotone* if $a \leq b$ in $\mathbb{B}$ implies $f(a) \leq' f(b)$ in $\mathbb{B}'$, *normal* if $f(\bot) = \bot'$, and *additive* if[2] $f(a \vee b) = f(a) \vee' f(b)$, and *multiplicative* if $f(a \wedge b) = f(a) \wedge' f(b)$. We will call an operation $f : B^n \to B$ an *operator* if it is normal and additive in each of its coordinates.

BAE_τ denotes the class of τ*-expanded Boolean algebra*, (shortly, τ-BAEs), that is, of algebras

$$\mathbb{A} = \langle A, \top, \bot, -, \wedge, \vee, \{\nabla^{\mathbb{A}} \mid \nabla \in \tau\} \rangle$$

with a Boolean *reduct* $\langle A, \top, \bot, -, \wedge, \vee \rangle$ that is indeed a Boolean algebra. $\mathbb{A}$ is called a *monotone* τ-expanded Boolean algebra, or a τ-BAM, if each $\nabla^{\mathbb{A}}$ is a monotone operation, and a *Boolean algebra with* τ*-operators*, or τ-BAO, if each $\nabla^{\mathbb{A}}$ is an operator. The classes of these algebras are denoted as, respectively, BAM_τ and BAO_τ. In the case of the basic modal similarity type, we speak of *modal* algebras rather than of τ-BAOs; MA denotes the class of these algebras. Given a set Γ of modal τ-formulas, and a class K of τ-expanded Boolean algebras, we define $\mathsf{K}(\Gamma)$ as the class of algebras in K that validate the set of equations $\Gamma^{\approx} := \{\gamma \approx \top \mid \gamma \in \Gamma\}$.

Given two τ-BAEs $\mathbb{A}$ and $\mathbb{A}'$, we call a map $\eta : A \to A'$ a *Boolean homomorphism* if it is a homomorphism from the Boolean reduct of $\mathbb{A}$ to that of $\mathbb{A}'$, and a *modal homomorphism* it if is a homomorphism with respect to the modal operations. Thus a homomorphism between two τ-BAEs is a map that is both a Boolean and a modal homomorphism. We let BAE_τ, BAM_τ, etc. also denote the category with the τ-BAEs, ..., as objects and the homomorphisms as arrows.

EXAMPLE 16. Algebras of the form $\mathbb{S}^+$, with $\mathbb{S}$ some τ-frame, are the prime specimens of Boolean algebras with operators. These algebras are sometimes referred to as *concrete* BAOs.

More generally, the complex algebra of a neighborhood frame (see Remark 10) is an example of a BAE for the basic modal similarity type; it is easy to see that such an $\mathbb{S}^+$ belongs to BAM iff $\mathbb{S}$ is a monotone neighborhood frame.

Our terminological convention will be that properties of and notions pertaining to Boolean algebras (such as atomicity, completeness, filters, ...) apply to an expanded Boolean algebra as they apply to its underlying Boolean algebras.

All of the properties defined in Definition 15 can be given in equational form, so all of the classes defined there are in fact *varieties*. In the next section we discuss the algebraic properties of these varieties; let us first see why they are so important from a logical perspective. This can be formulated very concisely.

THEOREM 17. *Let* Γ *be a set of modal* τ*-formulas. Then* $\mathsf{BAE}_\tau(\Gamma)$ *algebraizes* $\mathbf{C}_\tau.\Gamma$, $\mathsf{BAM}_\tau(\Gamma)$ *algebraizes* $\mathbf{M}_\tau.\Gamma$, *and* $\mathsf{BAO}_\tau(\Gamma)$ *algebraizes* $\mathbf{K}_\tau.\Gamma$. *In particular,* $\mathsf{MA}(\Gamma)$ *algebraizes* $\mathbf{K}.\Gamma$.

[2]Observe that we write $\vee$ and $\vee'$ rather than $\vee^{\mathbb{A}}$ and $\vee^{\mathbb{A}'}$, respectively; this convention will always apply to the interpretations of the Boolean symbols, and sometimes to the modal connectives as well.

Note that this theorem implies a general, algebraic, *completeness* result: for instance, concerning modal logics in the basic modal similarity type, it states that

$$\vdash_{\mathbf{K}.\Gamma} \varphi \text{ iff } \mathsf{MA}(\Gamma) \models \varphi^{\approx}. \tag{7}$$

That is to say, φ is a theorem of the logic *axiomatized* by Γ if and only if φ is valid in the class of algebras *defined* by Γ.

The key tool in the *proof* of Theorem 17 is played by the so-called *Lindenbaum-Tarski algebra* of a logic. The introduction of this fundamental tool is based on the observation that for all classical modal logics, the notion of logical equivalence is a congruence on the formula algebra.

DEFINITION 18. Let L be a modal τ-logic. The relation $\equiv_L$ between formulas is defined by putting $\varphi \equiv_L \psi$ if $\varphi \leftrightarrow \psi$ is an L-theorem.

PROPOSITION 19. *For any classical modal τ-logic L, the relation $\equiv_L$ is a congruence on the formula algebra $\mathbb{F}\mathsf{ma}_\tau$.*

DEFINITION 20. Given a modal τ-logic L, we denote the congruence class of the formula χ under the relation $\equiv_L$ by $[\chi]_L$; for a set of formulas Φ, we let $[\Phi]_L$ denote the set $\{[\varphi]_L \mid \varphi \in \Phi\}$. The quotient algebra $\mathbb{F}\mathsf{ma}_\tau/{\equiv_L}$ is called the *Lindenbaum-Tarski* algebra of L, notation: $\mathbb{F}_L$.

Note that the elements of the Lindenbaum-Tarski algebra $\mathbb{F}_L$ are the equivalence classes of the relation $\equiv_L$ of the set Fma_τ. The algebraic operations are defined as follows: $\top^{\mathbb{F}_L} = [\top]_L$, $\perp^{\mathbb{F}_L} = [\perp]_L$, $[\varphi]_L \wedge^{\mathbb{F}_L} [\psi]_L = [\varphi \wedge \psi]_L$, etc. We briefly remind the reader that all of these definitions could be parameterized by making the set X of variables explicit.

It is hard to overestimate the importance of Lindenbaum-Tarski algebras. For a start, the algebra $\mathbb{F}_L$ contains all the information of its logic L, in the following sense.

THEOREM 21. *Let L be a modal logic for some similarity type τ. Then for any two τ-formulas φ and ψ, we have*

$$\mathbb{F}_L \models \varphi \approx \psi \textit{ iff } \varphi \equiv_L \psi.$$

Proof. For the direction from left to right, consider the natural assignment $\nu : x \mapsto [x]_L$. It follows from the validity of $\varphi \approx \psi$ in $\mathbb{F}_L$ that $\widetilde{\nu}(\varphi) = \widetilde{\nu}(\psi)$. But an easy formula induction shows that $\widetilde{\nu}(\chi) = [\chi]_L$, for all formulas χ. Hence we obtain that $[\varphi]_L = [\psi]_L$, that is, $\varphi \equiv_L \psi$.

For the reverse direction, let α be some assignment on the Lindenbaum-Tarski algebra. Choose for each variable x a representative $\sigma(x)$ of the equivalence class $\alpha(x)$; that is, for each variable x we have that $\alpha(x) = [\sigma(x)]_L$. Note that this map σ is nothing but a substitution; recall that $\widetilde{\sigma}$ is the extension of σ to all formulas. It is not hard to prove that all formulas χ satisfy $\widetilde{\alpha}(\chi) = [\widetilde{\sigma}(\chi)]_L$. But it follows from $\varphi \equiv_L \psi$ that $\widetilde{\sigma}(\varphi) \equiv_L \widetilde{\sigma}(\psi)$, since L is closed under uniform substitution. Hence we find that $\widetilde{\alpha}(\varphi) = \widetilde{\alpha}(\psi)$. And since α was arbitrary, this shows that $\mathbb{F}_L \models \varphi \approx \psi$, as required. ❑

On the other hand, Lindenbaum-Tarski algebras play an important algebraic role as well, as is concisely formulated in the following Theorem.

THEOREM 22. *For any classical modal τ-logic L, $\mathbb{F}_L$ is the ω-generated free algebra for the variety $\mathsf{BAE}_\tau(L)$.*

Proof. Let $\mathbb{A}$ be an algebra in $\mathsf{BAE}_\tau(L)$, and consider an arbitrary map $\alpha : [X]_L \to A$ (recall that X denotes the set of variables, and that $[X]_L = \{[x]_L \mid x \in X\}$). We will prove that α can be extended to a homomorphism from $\mathbb{F}_L$ to $\mathbb{A}$.

To this aim, consider the composition $\alpha \circ \nu : X \to A$ of α with the natural map $\nu : x \mapsto [x]_L$. It follows from the universal mapping property of $\mathbb{F}\mathsf{ma}_\tau$ over X that this map can be extended to a homomorphism $\widetilde{\alpha \circ \nu} : \mathbb{F}\mathsf{ma}_\tau \to \mathbb{A}$.

We claim that $\ker(\widetilde{\nu}) \subseteq \ker(\widetilde{\alpha \circ \nu})$. To see this, consider formulas φ and ψ such that $(\varphi, \psi) \in \ker(\nu)$; then $[\varphi]_L = [\psi]_L$, and so $\varphi \equiv_L \psi$. It follows from $\mathbb{A}$ being in $\mathsf{BAE}_\tau(L)$ that $\mathbb{A} \models \varphi \approx \psi$, so $\varphi \approx \psi$ certainly holds in $\mathbb{A}$ under the assignment $\alpha \circ \nu$. But that is just another way of saying that $(\varphi, \psi) \in \ker(\widetilde{\alpha \circ \nu})$.

But then from this claim it follows that the map $\widetilde{\alpha} : Fma_\tau/{\equiv_L} \to A$, given by

$$\widetilde{\alpha}([\varphi]_L) := \widetilde{\alpha \circ \nu}(\varphi)$$

is well-defined. It is not hard to show that $\widetilde{\alpha}$ is in fact a homomorphism from $\mathbb{F}_L$ to $\mathbb{A}$, and since it clearly extends α, we have established the universal mapping property of $\mathbb{F}_L$ for $\mathsf{BAE}_\tau(L)$ over $[X]_L$. ❑

Finally, in order to prove the Algebraization Theorem 17 from these two theorems, we need one additional result concerning varieties of the form $\mathsf{BAE}_\tau(L)$ if L is a modal logic axiomatized by a set Γ of formulas. We leave the rather tedious but straightforward proof of this proposition as an exercise for the reader.

PROPOSITION 23. *Let* Γ *be a set of* τ*-formulas. Then* $\mathsf{BAE}_\tau(\mathbf{C}_\tau.\Gamma) = \mathsf{BAE}_\tau(\Gamma)$, $\mathsf{BAE}_\tau(\mathbf{M}_\tau.\Gamma) = \mathsf{BAM}_\tau(\Gamma)$, *and* $\mathsf{BAE}_\tau(\mathbf{K}_\tau.\Gamma) = \mathsf{BAO}_\tau(\Gamma)$.

This finishes our introduction to the algebraization of modal logics. In section 6 we will have a lot more to say about the link between normal modal logics and varieties of BAOs.

4 VARIETIES OF EXPANDED BOOLEAN ALGEBRAS

In this section we discuss what the theory of universal algebra has to say about Boolean algebras with operators and their siblings.

Lattices of congruences

A very important theme in universal algebra has been to relate the properties of a variety to the shape of the congruence lattices of its algebras. In the case of Boolean algebras and their expansions, this has turned out to be particularly fruitful.

DEFINITION 24. An algebra $\mathbb{A}$ has *permuting congruences* if $\Theta_1 \circ \Theta_2 = \Theta_2 \circ \Theta_1$ for all congruences Θ_1, Θ_2 over $\mathbb{A}$; $\mathbb{A}$ is *congruence distributive* if $\mathbb{C}\mathsf{g}(\mathbb{A})$, its lattice of congruences, is distributive.

These properties hold of a variety if they hold of each of its members; and a variety is called *arithmetical* if it is both congruence permutable and congruence distributive.

It is a rather strong property for an algebra to have permuting congruences, or to be congruence distributive, and both notions have important applications. Concerning the second notion, we will see an important property of congruence distributive varieties

in Theorem 35. In order to motivate the first concept here we just mention that it allows a considerable simplification in the computation of joins in congruence lattices: whereas in general the join $\Theta_1 \vee \Theta_2$ of two congruences Θ_1 and Θ_2 is given as $\Theta_1 \vee \Theta_2 = \Theta_1 \ \cup \ (\Theta_1 \circ \Theta_2) \ \cup \ (\Theta_1 \circ \Theta_2 \circ \Theta_1) \ \cup \ \cdots$, in the case of permuting congruence this rearranges itself as $\Theta_1 \vee \Theta_2 = \Theta_1 \circ \Theta_2$.

THEOREM 25. *Varieties of expanded Boolean algebras are arithmetical.*

Proof. This proof can be seen as a consequence of a result by A. Pixley, who proved that a variety is arithmetical if and only if it admits the definition of so-called *Mal'cev* and $\frac{2}{3}$*-majority terms.* For some detail, let V be a variety of expanded Boolean algebras. First consider the ternary (Boolean) term $p(x, y, z)$ given by

$$p(x, y, z) := (x \wedge z) \vee (x \wedge \neg y \wedge \neg z) \vee (\neg x \wedge \neg y \wedge z).$$

We leave it for the reader to verify that this is a *Mal'cev* term, that is,

$$\mathsf{V} \models p(x, x, z) \approx z \text{ and } \mathsf{V} \models p(x, z, z) \approx x. \tag{8}$$

From this it follows that V is congruence permutable: let $\mathbb{A}$ be some algebra in the variety and let $a, b \in A$ be elements such that $(a, b) \in \Theta_1 \circ \Theta_2$ for some congruences Θ_1 and Θ_2. Then there is some $c \in A$ with $(a, c) \in \Theta_1$ and $(c, b) \in \Theta_2$. From this it follows that $(a, b) \in \Theta_2 \circ \Theta_1$, because

$$a = p^{\mathbb{A}}(a, b, b)\Theta_2 p^{\mathbb{A}}(a, c, b)\Theta_1 p^{\mathbb{A}}(c, c, b) = b.$$

This proves that $\Theta_1 \circ \Theta_2 \subseteq \Theta_2 \circ \Theta_1$ which means that $\mathbb{A}$ has permuting congruences. Congruence distributivity can be proven in a similar way: consider the term M given by

$$M(x, y, z) := (x \vee y) \wedge (y \vee z) \wedge (z \vee x).$$

The reader will have little trouble in showing that

$$\mathsf{V} \models M(x, x, y) \approx M(x, y, x) \approx M(y, x, x) \approx x, \tag{9}$$

i.e., M is a $\frac{2}{3}$*-majority term.* In a similar way as above we can then use (9) to show V is congruence distributive. ❑

Congruences and filters

One of the nicest features of BAOs is that their congruences can be characterized by certain *subsets* of the algebra.

DEFINITION 26. Let $\mathbb{B}$ be a Boolean algebra. A subset $F \subseteq B$ is called a *(Boolean) filter* of $\mathbb{B}$ if it (i) contains the top element of $\mathbb{B}$, (ii) is closed under taking meets (that is, if $a, b \in F$ then $a \wedge b \in F$), and (iii) is an *up-set* (that is, $a \in F$ and $a \leq b$ imply $b \in F$). A filter F is *proper* if it does not contain the bottom element of $\mathbb{B}$, or equivalently, if $F \neq B$. We let $Fi(\mathbb{B})$ denote the collection of filters of $\mathbb{B}$.

EXAMPLE 27. It is not difficult to see that the set $Fi(\mathbb{B})$ is closed under taking intersections; hence, we may speak of the *smallest* filter F_D containing a given set $D \subset B$; this filter can also be defined as the following set

$$F_D = \{\top\} \cup \{b \in B \mid \text{ there are } d_1, \ldots, d_n \in D \text{ such that } d_1 \wedge \cdots \wedge d_n \leq b\},$$

which explains why we also refer to this set as the filter *generated by* D. In case that D is a singleton $\{a\}$, we write $a\!\uparrow$ for $F_{\{a\}}$; this set is called the *principal filter generated by* a. Clearly we have $a\!\uparrow := \{b \in \mathbb{B} \mid a \leq b\}$.

The filter F_D is proper iff D has the so-called *finite meet property* (that is, $\bigwedge D_0 > \bot$ for all finite subsets $D_0 \subseteq D$).

DEFINITION 28. Let $\mathbb{A}$ be a BAO; a subset $F \subseteq A$ is a *modal* or *open filter* of $\mathbb{A}$ if F is a filter of (the underlying BA of) $\mathbb{A}$ which is closed under the application of boxes; that is, $a \in F$ implies $\Box_i a \in F$ for all boxes $\Box_i$. (If the language has modalities of arity higher > 1, we need to strengthen this to requiring that F is closed under the application of induced boxes.) The collection of modal filters of $\mathbb{A}$ is denoted as $MFi(\mathbb{A})$.

In any BAO $\mathbb{A}$, the sets $\{\top^{\mathbb{A}}\}$ and A are modal filters; the singleton $\{\top^{\mathbb{A}}\}$ is called the *trivial* (modal) filter of $\mathbb{A}$, and any filter different from A is called *proper*.

The following theorem will prove to be extremely useful.

THEOREM 29. *Let $\mathbb{A}$ be a Boolean algebra with operators. Then*

1. *the collection $MFi(\mathbb{A})$ is closed under taking arbitrary intersections and hence forms a complete lattice with respect to the subset ordering;*

2. *this lattice is isomorphic to the congruence lattice of $\mathbb{A}$ through the isomorphism $\Pi : MFi(\mathbb{A}) \to Cg(\mathbb{A})$ given by*

$$\Pi_M := \{(a, b) \in A \times A \mid a \leftrightarrow b \in M\},$$

and its inverse $N : Cg(\mathbb{A}) \to MFi(\mathbb{A})$ by

$$N_\Theta := \{a \in A \mid (a, \top) \in \Theta\}.$$

It follows from the completeness of the lattice of modal filters of a BAO $\mathbb{A}$, that with each set $D \subseteq A$ we may associate the *smallest* modal filter M_D including D. The following proposition explains why we also refer to M_D as the modal filter *generated by* D:

PROPOSITION 30. *Let $\mathbb{A}$ be a Boolean algebra with τ-operators, and D a subset of A. Then*

$$M_D = \{a \in A \mid \blacksquare_1 d_1 \wedge \ldots \wedge \blacksquare_n d_n \leq a \textit{ for some } \blacksquare_1, \ldots, \blacksquare_n \in CD(\tau),\ d_1, \ldots, d_n \in D\}.$$

In particular, when D is a singleton, say, $D = \{d\}$, we find

$$M_d = \{a \in A \mid \blacksquare d \leq a \textit{ for some } \blacksquare \in CD(\tau),\ d \in D\}.$$

Subdirect irreducibility

We now turn to the algebraic notion of subdirect irreducibility, which plays an important role in the analysis of varieties. The motivation for introducing this concept is the search for the universal algebraic analogon of the prime numbers, as it were. That is, we want to isolate a class of algebraic building blocks that are basic in the sense that (i) every algebra may be decomposed into basic ones, while (ii) the basic ones themselves only

allow trivial decompositions. Now there are various interpretations of the words 'basic' and 'decomposition'.

DEFINITION 31. An algebra $\mathbb{A}$ is *simple* if its only homomorphic images are $\mathbb{A}$ itself and the trivial algebra of its similarity type, and *directly indecomposable* if in any decomposition $\mathbb{A} \cong \prod \mathbb{A}_i$, $\mathbb{A}$ is isomorphic to one of the $\mathbb{A}_i$.

Both of these notions are important and interesting, but neither one is exactly what we want. The notion of simplicity is too restrictive since not every variety is generated by its simple members. And, whereas every finite algebra is isomorphic to a direct product of directly indecomposable algebras, this does not hold for all infinite algebras. For instance, it is not hard to see that the algebra $\mathbb{2}$ of Example 8 is the only nontrivial directly indecomposable Boolean algebra, while a straightforward cardinality argument shows that no countably infinite algebra can be isomorphic to a direct power of $\mathbb{2}$.

Hence, in order to meet our criteria, we arrive at a notion which at first sight may seem somewhat involved. In words, an algebra is subdirectly irreducible iff it does not allow a proper *subdirect* decomposition.

DEFINITION 32. Let $\mathbb{A}$ be an algebra, and $\{\mathbb{A}_i \mid i \in I\}$ a family of algebras of the same type. An embedding η of $\mathbb{A}$ into $\prod_{i\in I} \mathbb{A}_i$ is called *subdirect* if $\pi_i \circ \eta$ is surjective for each projection function π_i. If $\mathbb{A}$ is a subalgebra of $\prod_{i\in I} \mathbb{A}_i$, then we say that $\mathbb{A}$ is a *subdirect product of* the family $\{\mathbb{A}_i \mid i \in I\}$, or that the family forms a *subdirect decomposition of* $\mathbb{A}$, if the inclusion map is a subdirect embedding.

$\mathbb{A}$ is called *subdirectly irreducible*, or, briefly, *s.i.*, if for every subdirect embedding $\eta : \mathbb{A} \to \prod_{i\in I} \mathbb{A}_i$ there is an $i \in I$ such that $\pi_i \circ \eta : \mathbb{A} \to \mathbb{A}_i$ is an isomorphism.

In practice, one always uses a nice characterization of subdirect irreducibility in terms of the congruence lattice of the algebra, and similarly for simple and directly indecomposable algebras. For the proof of this proposition we refer to any standard textbook on universal algebra. For a proper understanding of its formulation, recall that any algebra $\mathbb{A}$ always has at least two congruences: the *diagonal* relation $\Delta_A = \{(a, a) \mid a \in A\}$, and the *global* relation $\Upsilon_A = A \times A$.

PROPOSITION 33. *Let $\mathbb{A}$ be an algebra. Then*

1. *$\mathbb{A}$ is simple iff $Cg(\mathbb{A}) = \{\Delta, \Upsilon\}$;*
2. *$\mathbb{A}$ is directly indecomposable iff there are no two congruences Θ_1 and Θ_2 such that $\Theta_1 \wedge \Theta_2 = \Delta$ and $\Theta_1 \circ \Theta_2 = \Upsilon$;*
3. *$\mathbb{A}$ is subdirectly irreducible iff it has a monolith, that is, a smallest non-diagonal congruence.*

The following theorem can be read as stating that, indeed, subdirect irreducibility is the proper concept when it comes to finding the basic building blocks of varieties.

THEOREM 34 (Birkhoff). *Every algebra can be subdirectly decomposed into subdirectly irreducible algebras. As a corollary, every variety is generated by its subdirectly irreducible members.*

As a corollary of this theorem, we see that the study of the lattice of subvarieties of a given variety can be conducted by way of inspecting the s.i. members of the variety. In the case of expanded Boolean algebras, the logical meaning of this is that it gives us a

tool for the study of extensions of a given modal logic. For, as we will see in section 6 that the subvarieties of the variety determined by a modal logic, correspond to the extensions of that logic. Also, because expanded Boolean algebras are congruence distributive, we may apply *Jónsson's Lemma*. This result involves the class operations H, S and Pu, which are defined in the appendix.

THEOREM 35 (Jónsson). *Let K be a class of algebras such that $\mathsf{Var}(\mathsf{K})$ is congruence distributive. Then all subdirectly irreducible members of $\mathsf{Var}(\mathsf{K})$ belong to $\mathsf{HSPu}(\mathsf{K})$.*

The use of this theorem lies in the fact that if K generates a congruence distributive variety V, then the s.i. members of V still resemble the algebras in K in many ways. For instance, if K is a *finite* set of *finite* members, then $\mathsf{Pu}(\mathsf{K}) = \mathsf{K}$; hence we obtain the following result for finitely generated varieties of expanded Boolean algebras.

COROLLARY 36. *Let K be a finite set of finite τ-expanded Boolean algebras. Then $\mathsf{Var}(\mathsf{K})$ only has finitely many subvarieties, each of which is determined by a subset of $\mathsf{HS}(\mathsf{K})$.*

Finally, restricting our attention to Boolean algebras with operators, we encounter yet another nice property, namely that we can characterize subdirect irreducibility of an algebra by the existence of one *single* element — one with rather special properties, that is.

DEFINITION 37. An element e of a BAO $\mathbb{A}$ is called *essential* or an *opremum* if $e < \top$, while for all $b < \top$ there is a compound modality $\blacklozenge$ such that $\blacksquare b \leq e$. Dually, we say that an element ρ is *radical in* $\mathbb{A}$, or a *radix of* $\mathbb{A}$, if $\rho > \bot$, while for all $a > \bot$ there is a compound modality $\blacklozenge$ such that $\rho \leq \blacklozenge a$.

Clearly, an element e of a BAO is essential iff its complement $-e$ is radical. In the sequel this fact will be used implicitly, context deciding which formulation is the most convenient.

EXAMPLE 38. Let $\mathbb{S}$ be a *rooted* frame with root r. It is easy to see that the singleton $\{r\}$ is radical in $\mathbb{S}^+$: let $a \subseteq S$ be a nonempty element of $\mathbb{S}^+$. Take an element s from a; since r is a root of $\mathbb{S}$, there must be some compound modality $\blacklozenge$ such that $R_\blacklozenge rs$; from this it is immediate that $\{r\} \subseteq \langle R_\blacklozenge \rangle a$.

The following theorem (or at least, the more important statement concerning subdirect irreducibility) is due to Rautenberg, see for instance [91].

THEOREM 39. *Let $\mathbb{A}$ be a nontrivial Boolean algebra with τ-operators. Then $\mathbb{A}$ is simple iff every non-top element of A is essential, and subdirectly irreducible iff it has an essential element.*

Proof. It follows immediately from Theorem 29 that $\mathbb{A}$ is s.i. iff it has a smallest non-trivial modal filter, and it is not hard to see that any such filter is of the form M_e for some element e of $\mathbb{A}$. The proof of the statement on subdirect irreducibility is thus complete if we can show that for an arbitrary element $e \in A$:

$$M_e \text{ is a smallest nontrivial modal filter iff } e \text{ is essential.} \tag{10}$$

First suppose that M_e is a smallest nontrivial modal filter. Since M_e is nontrivial, it follows immediately that $e \neq \top$. In order to show that e is essential, consider an arbitrary element $a < \top \in A$, and consider the filter M_a generated by a. It follows from

our assumption on M_e that $M_e \subseteq M_a$, so that $e \in M_a$. Hence we may deduce from Proposition 30 that there is some compound modality $\blacklozenge$ such that $\blacksquare a \leq e$. This suffices to prove that e is essential.

For the converse direction, suppose that e is essential, and let M be an arbitrary nontrivial modal filter on $\mathbb{A}$. That is, $M \neq \{\top\}$, so M contains an element $a \neq \top$. but then it follows from the essentiality of e that there is some compound modality $\blacklozenge$ such that $\blacksquare a \leq e$; this shows that $e \in M$, whence $M_e \subseteq M$. In other words, M_e is the smallest modal filter on $\mathbb{A}$.

The proof concerning simplicity is completely similar and therefore left as an exercise. ❑

5 FRAMES AND ALGEBRAS

5.1 *Introduction*

The algebraic study of modal logic was started in section 3. Its main result, Theorem 17, links normal modal logics to varieties of Boolean algebras with operators by stating a general algebraization result. But no matter how well-behaved these algebras are, most modal logicians will still prefer the relational semantics, either because they find it more intuitive, or because frames simply happen to be the structures in which they take an (application driven) interest. Hence there is an obvious need to understand the precise relation between the worlds of frames and algebras, respectively. As we will discuss in this section, much of this relation can be understood within the framework of two dualities, both of which relate algebras to (topological) frames, and one forgetful functor. In order to explain why *two* dualities are needed, it is best to consider *finite* structures first. For the sake of a smooth presentation we confine ourselves to the basic modal language.

Let $\mathsf{FinFram}$ and FinMA denote the respective categories of finite frames with bounded morphisms, and of finite modal algebras with homomorphisms. Recall that in Definition 9 we coded up a frame $\mathbb{S} = (S, R)$ by means of its *complex algebra* $\mathbb{S}^+$. Conversely, if $\mathbb{A} = \langle A, \perp, \top, -, \wedge, \vee, \diamond \rangle$ is a finite modal algebra, then we can base a frame on the set $At(\mathbb{A})$ of *atoms* (see Definition 40) of $\mathbb{A}$ by putting

$$R_\diamond pq \ :\Longleftrightarrow \ \ p \leq \diamond q.$$

It is then easy to see that

$$\mathbb{S} \cong (\mathbb{S}^+)_+ \text{ and } \mathbb{A} \cong (\mathbb{A}_+)^+$$

for an arbitrary finite frame $\mathbb{S}$ and an arbitrary finite modal algebra $\mathbb{A}$. And, with the appropriate extension of the constructions $(\cdot)^+$ and $(\cdot)_+$ to functors, we can in fact establish that

$$(\cdot)^+ \text{ and } (\cdot)_+ \text{ form a dual equivalence between } \mathsf{FinFram} \text{ and } \mathsf{FinMA}. \tag{11}$$

Unfortunately, there is no way to remove the restriction to finite structures in (11) and obtain a dual equivalence between the categories Fr and MA. In fact, since the category MA has an initial object (the free modal algebra over zero generators), while Fr does not have a final object (cf. section 10 for details), *no* duality whatsoever can be established between these two categories. However, there *is* a natural way to associate a frame with an arbitrary modal algebra $\mathbb{A}$, if we let *ultrafilters* generalize the notion of an atom. That

is, we can simply base the *ultrafilter frame* $\mathbb{A}_\bullet$ of $\mathbb{A}$ on the collection of ultrafilters of (the Boolean reduct of) $\mathbb{A}$ by putting

$$R_\Diamond uv \;:\Longleftrightarrow\; \Diamond a \in u \text{ for all } a \in v.$$

Again, this construction can be extended to a functor $(\cdot)_\bullet$ from MA to Fr.

We will see that there is interesting interaction between the functors $(\cdot)^+$ and $(\cdot)_\bullet$. The most important result is the Jónsson-Tarski representation theorem stating that every modal algebra $\mathbb{A}$ can be embedded in its 'double dual' $\mathbb{A}^\sigma := (\mathbb{A}_\bullet)^+$. As we will see in the next section, this result lies at the root of the application of algebra in modal completeness results.

While there is no duality between the categories Fr and MA, with some modifications, both functors $(\cdot)^+$ and $(\cdot)_\bullet$ do provide interesting dualities. Here there are two basic observations. First, the complex duality functor $(\cdot)^+$ is injective on objects; that is, any frame may be recovered (modulo isomorphism) from its complex algebra. Second, although the functor $(\cdot)_\bullet$ does not have this property (sse Example 53), there is a simple remedy for this problem, namely, to *add* the missing information, topologically encoded, to the frame $\mathbb{A}_\bullet$ of an algebra $\mathbb{A}$. Thus we see that two fairly nice dualities can be found if we remove the finiteness constraint on *either* side of the duality (11):

- a 'complex' or 'discrete' duality obtains (see Theorem 47) if we consider the entire category on the frame side, and a *subcategory* of *perfect* algebras with *complete* homomorphisms on the other side;
- a 'topological' duality obtains (see Theorem 67) if, conversely, we keep the category on the algebra side intact, but *add topological structure* on the frame side.

Both dualities restrict to (11) in the finite case, and the topological and the complex duality are linked by the functor that *forgets* the topological structure on the frame side. Furthermore, similar results can be proved connecting (monotone) neighborhood frames and (monotone) expanded Boolean algebras. In fact, the picture sketched above applies to far wider contexts [68].

For a brief overview of this section, below we first introduce the above mentioned functors and dualities, in some detail. We then see how the algebraic notions of subdirect irreducibility and simplicity turn up on the other side of this duality. We finish the section with a brief discussion of the interaction of the functors $(\cdot)^+$ and $(\cdot)_\bullet$ with more 'intrinsic' constructions on algebras and frames such as products and disjoint unions.

5.2 Complex duality

We have already seen how to transform frames into algebras; we now consider these complex algebras from a more abstract perspective. In order to characterize them among the class of all Boolean algebras with operators, we need some terminology.

DEFINITION 40. A Boolean algebra $\mathbb{B}$ is called *complete* if it is complete as a lattice, that is, if every subset X of B has both a meet (or greatest lower bound) $\bigwedge X$ and a join (or least upper bound) $\bigvee X$. $\mathbb{B}$ is called *atomic* if below every non-bottom element of $\mathbb{B}$ there is an *atom*, (i.e., an element p satisfying $\bot < p$ while there is no a such that $\bot < a < p$).

Now let $\mathbb{B}$ and $\mathbb{B}'$ be two Boolean algebras; a map $f : B \to B'$ is called *completely additive* if it preserves all non-empty joins, that is, if for all non-empty subsets X of B for which $\bigvee X$ exists, it holds that

$$f(\bigvee X) = \bigvee' f[X].$$

An n-ary operation f on a Boolean algebra $\mathbb{B}$ is called a *complete operator* if it preserves all joins in each coordinate (or, equivalently, if it is normal and completely additive in each of its coordinates). Finally, a Boolean algebra with operators is called *perfect* if it is complete and atomic, and all its operators are complete.

The reader can easily verify that all complex algebras are perfect. It is equally easy to see that every finite BAO is perfect, since such an algebra has no infinite joins, and a straightforward induction proves that operators preserve finite joins in each of their arguments. For an example of an operator that is not complete, let S be an infinite set, and define $f : \mathcal{P}(S) \to \mathcal{P}(S)$ by putting $f(X) = X$ if X is finite while $f(X) = S$ otherwise.

In the very same way as we defined above for finite structures, given a perfect BAO we can define a frame based on the set of atoms of $\mathbb{A}$. In fact, for the definition to make sense, we only need the BAO to be atomic.

DEFINITION 41. Let $\mathbb{B}$ be an atomic Boolean algebra, and f an n-ary operator on $\mathbb{B}$. Define the $n+1$-ary relation Q_f on $At(\mathbb{B})$ by

$$Q_f p_0 p_1 \dots p_n \;:\Longleftrightarrow\; p_0 \leq f(p_1, \dots, p_n).$$

Given an atomic τ-BAO $\mathbb{A}$, define its *atom structure* $\mathbb{A}_+$ as the τ-frame $\mathbb{A}_+ = \langle At\mathbb{A}, \{Q_{\nabla^{\mathbb{A}}} \mid \nabla \in \tau\}\rangle$.

Now that we have ways to turn frames into atomic algebras and vice versa, the natural question is how these constructions interact. The following proposition seems to be folklore.

PROPOSITION 42. *Let, for a modal similarity type τ, $\mathbb{S}$ be a τ-frame, and $\mathbb{A}$ an atomic τ-BAO. Then*

1. $\mathbb{S} \cong (\mathbb{S}^+)_+$;
2. $\mathbb{A} \cong (\mathbb{A}_+)^+$ *iff* $\mathbb{A}$ *is perfect.*

Proof. Concerning the first part, it is straightforward to verify that the map $\iota : x \mapsto \{x\}$ is the required isomorphism. For the second item, let the map $\epsilon : A \to \mathcal{P}(At(\mathbb{A}))$ be given by $\epsilon(a) := \{p \in At\mathbb{A} \mid p \leq a\}$. The crucial observation in the proof is that

$$\epsilon \text{ embeds } \mathbb{A} \text{ into } (\mathbb{A}_+)^+ \text{ iff all operations of } \mathbb{A} \text{ are complete.} \qquad (12)$$

This map is then an isomorphism iff $\mathbb{A}$ is perfect. ❑

As we will see now, the link between frames and algebras is not restricted to objects. With the natural definition for morphisms between perfect BAOs, we will see how to turn bounded morphisms between frames into these *complete* BAO homomorphisms, and vice versa.

DEFINITION 43. Let $\mathbb{A}$ and $\mathbb{A}'$ be two perfect τ-BAOs. A *complete homomorphism* from $\mathbb{A}$ to $\mathbb{A}'$ is a homomorphism $\eta : \mathbb{A} \to \mathbb{A}'$ which preserves *all* meets and joins. That is, for every subset $X \subseteq A$ we have that $\eta(\bigvee X) = \bigvee' \eta[X]$ and $\eta(\bigwedge X) = \bigwedge' \eta[X]$. We let BAO_τ^+ denote the category of perfect Boolean algebras with τ-operators as objects, and complete homomorphisms as arrows.

DEFINITION 44. Let $\mathbb{S}$ and $\mathbb{S}'$ two τ-frames. Given a bounded morphism $\theta : \mathbb{S} \to \mathbb{S}'$, define the map $\theta^+ : \mathcal{P}(S') \to \mathcal{P}(S)$ by

$$\theta^+(X') := \{s \in S \mid \theta(s) \in X'\}.$$

Conversely, given perfect τ-BAOs $\mathbb{A}$ and $\mathbb{A}'$ and a complete homomorphism $\eta : \mathbb{A} \to \mathbb{A}'$, define the map $\eta_+ : At(\mathbb{A}') \to A$, which can be shown to map atoms to atoms, by

$$\eta_+(p') := \bigwedge\{a \in A \mid p' \leq \eta(a)\}.$$

It is our aim to prove that $(\cdot)^+$ and $(\cdot)_+$ form a duality between the categories Fr_τ and BAO_τ^+. We first show functoriality:

PROPOSITION 45. *$(\cdot)^+$ is a contravariant functor from Fr_τ to BAO_τ^+.*

Proof. The important issue here is that for any bounded morphism $\theta : \mathbb{S} \to \mathbb{S}'$, the map θ^+ is a complete homomorphism from $\mathbb{S}'^+$ to $\mathbb{S}^+$. It is easy to see that θ^+ is a complete Boolean homomorphism between the respective power set algebras; in order to prove that it is also a modal homomorphism, it suffices to show that for an $n+1$-ary relation R we have

$$\langle R\rangle(\theta^+(X_1), \ldots, \theta^+(X_n)) = \theta^+\left(\langle R'\rangle(X_1, \ldots, X_n)\right) \tag{13}$$

in case θ is a bounded morphism with respect to R and R'. Here it is interesting to note that in fact the inclusion $\subseteq$ is equivalent to the *forth property*, and the converse inclusion $\supseteq$, to the *back property* of θ. In a way, (13) can be seen as a piece of evidence that bounded morphisms provide in fact the right kind of morphism between frames. ❑

PROPOSITION 46. *$(\cdot)_+$ is a contravariant functor from BAO_τ^+ to Fr_τ.*

Proof. Here the first point is to prove that if $\eta : \mathbb{A} \to \mathbb{A}'$ is a complete Boolean homomorphism between the perfect τ-BAOs $\mathbb{A}$ and $\mathbb{A}'$, then η_+ maps atoms of $\mathbb{A}'$ to atoms of $\mathbb{A}$. To see this, let p' be an atom of $\mathbb{A}'$; it suffices to show that $\eta_+(p')$ is *join prime* in $\mathbb{A}$. That is, we assume that $\eta_+(p') \leq \bigvee X$ for some $X \subseteq A$, and have to show that $\eta_+(p') \leq x$ for some $x \in X$. From the assumption we may derive that

$$p' \leq' \eta\eta_+(p') \leq' \eta(\bigvee X) = \bigvee{}' \eta[X].$$

Here the first inequality directly follows from the definition of $\eta_+(p')$. But since p' is an atom of $\mathbb{A}'$, the fact that $p' \leq' \bigvee' \eta[X]$ implies that $p' \leq' \eta(x)$ for some $x \in X$. The definition of $\eta_+(p')$ then immediately gives that $\eta_+(p') \leq' x$.

Unfortunately, we do not have the space here to prove that if η is in addition a modal homomorphism, then η_+ is a bounded morphism, or that the operation $(\cdot)_+$ commutes with function composition, i.e., that $(\theta \circ \eta)_+ = \eta_+ \circ \theta_+$ if $\eta : \mathbb{A} \to \mathbb{A}'$ and $\theta : \mathbb{A}' \to \mathbb{A}''$ are complete homomorphisms. ❑

The following result, that we will refer to as the *complex duality* for BAOs, is due to Thomason [103] (for the basic modal logic case).

THEOREM 47. *The functors* $(\cdot)^+$ *and* $(\cdot)_+$ *constitute a dual equivalence between the categories* Fr_τ *and* BAO^+_τ.

Proof. Given the results already established, it suffices to prove that the isomorphisms $\iota_\mathbb{S} : \mathbb{S} \to (\mathbb{S}^+)_+$ and $\epsilon_\mathbb{A} : \mathbb{A} \to (\mathbb{A}_+)^+$, defined in the proof of Proposition 42, are *natural*. For instance, concerning ϵ, we have to prove that $\epsilon_{\mathbb{A}'} \circ \eta = (\eta_+)^+ \circ \epsilon_\mathbb{A}$ for an arbitrary complete homomorphism $\eta : \mathbb{A} \to \mathbb{A}'$. The reader can easily verify this by a direct calculation. ❑

5.3 *Ultrafilter frames*

Now let us see how to remove the restriction to finite structures on the algebra side of (11); our first goal is to represent arbitrary (that is, not necessarily finite or even atomic) algebras by frames. But, given a BAO $\mathbb{A}$, what to take as the points of a frame representing $\mathbb{A}$? This problem of course already appears on the Boolean level, and its solution is provided by Stone's representation theorem. This celebrated piece of mathematics states that every Boolean algebra can be embedded in the set algebra over its *ultrafilters*; let us briefly review the basic facts concerning ultrafilters.

DEFINITION 48. Let $\mathbb{B}$ be a Boolean algebra. An *ultrafilter* of $\mathbb{B}$ is a proper filter u such that either a or $-a$ belongs to u, for all $a \in B$. The collection of ultrafilters of $\mathbb{B}$ is denoted as $Uf(\mathbb{B})$. Given a set S, we sometimes refer to ultrafilters of the power set algebra of S as *ultrafilters over* S.

EXAMPLE 49. Given a set S, and an element $s \in S$, define the *principal* ultrafilter π_s as the set $\{X \subseteq S \mid s \in X\}$. It is straightforward to verify that this set is indeed an ultrafilter over S. More generally, if p is an atom of the BA $\mathbb{B}$, then the principal filter $p{\uparrow} = \{a \in B \mid p \leq a\}$ is in fact an ultrafilter; it is in this sense that ultrafilters form a generalization of atoms.

For an example of a non-principal ultrafilter, consider the Boolean algebra of finite and cofinite sets of some infinite set T; the collection of cofinite subsets of T forms an ultrafilter of this algebra.

As a last example, ultrafilters can be seen to generalize the notion of a *maximal consistent set*. Consider the Lindenbaum-Tarski algebra $\mathbb{F}_L$ of a modal logic L; it is easy to verify that Φ is a maximal L-consistent set of formulas if and only if the set $\{[\varphi]_L \mid \varphi \in \Phi\}$ is an ultrafilter of $\mathbb{F}_L$.

Ultrafilters can be characterized as the proper filters that are maximal with respect to the inclusion ordering; this identification provides the key tool for establishing the existence of ultrafilters, as the proof sketch of the following Theorem reveals.

THEOREM 50 (Ultrafilter Theorem). *Any proper filter of a Boolean algebra* $\mathbb{B}$ *can be extended to an ultrafilter of* $\mathbb{B}$.

Proof. Given a proper filter F, apply Zorn's Lemma to the collection C of proper filters that extend F, and obtain a proper filter u that is maximal in C. It is not hard to prove that u is in fact a maximal proper filter, and from this it is easily follows that u is an ultrafilter. ❑

Stone's representation theorem suggests to take the collection $Uf(\mathbb{A})$ of a BAO $\mathbb{A}$ as the *domain* of a frame that will represent $\mathbb{A}$; for the *accessibility relation* on this ultrafilter frame we will (in the case of the basic modal similarity type) make the ultrafilter v visible from u if there is no explicit information preventing this; that is, if there is no $a \in v$ with $\Diamond a \notin u$. For an arbitrary similarity type we have the following definition.

DEFINITION 51. Given an n-ary operator f on the Boolean algebra $\mathbb{B}$, define its *dual relation* R_f as the $n+1$-ary relation on $Uf(\mathbb{B})$ given by:

$$R_f uu_1 \cdots u_n \;:\Longleftrightarrow\; f(a_1, \ldots, a_n) \in u \text{ for all } a_1 \in u_1, \ldots, a_n \in u_n.$$

Now let $\mathbb{A}$ be a Boolean algebra with τ-operators; then we define the *ultrafilter frame* or *canonical structure* of $\mathbb{A}$ as the structure

$$\mathbb{A}_\bullet := \langle Uf(\mathbb{A}), \{R_{\nabla^\mathbb{A}} \mid \nabla \in \tau\}\rangle.$$

Given a class K of algebras, we let $\mathsf{Cst}(\mathsf{K})$ denote the class of ultrafilter frames of algebras in K.

EXAMPLE 52. Recall from Chapter 2 of this volume that the *canonical frame* of a normal modal logic L is the structure $\mathbb{C}_L = \langle C, R\rangle$ where C is the set of maximally L-consistent set of formulas, and (we confine ourselves to the basic modal similarity type) R is the canonical accessibility relation given by $Ruv \;:\Longleftrightarrow\; \Diamond\varphi \in u$ for all φ in v. Using the identification that we made in Example 49 of maximal L-consistent sets with ultrafilters of the Lindenbaum-Tarski algebra $\mathbb{F}_L$, it is fairly obvious that the canonical frame for L is isomorphic to the ultrafilter frame of $\mathbb{F}_L$.

As a second example of the ultrafilter frame construction we mention that the *ultrafilter extension* $\mathbf{ue}\mathbb{S}$ of a frame $\mathbb{S}$ (as defined in Chapter 5 of this volume) is nothing but the 'double dual' $(\mathbb{S}^+)_\bullet$ of $\mathbb{S}$. Verifying this is simply a matter of unraveling the definitions.

Unlike the complex algebra functor, the ultrafilter frame construction is not injective.

EXAMPLE 53. Let A be the collection of finite and cofinite subsets of $\mathbb{N}$, and let B contain in addition those sets of natural numbers that differ in at most finitely many elements from either the set E of evens or the set O of odds. Both A and B are closed under the Boolean operations, and it is easy to see that A has exactly one non-principal ultrafilter, and B, exactly two: one containing the set E, and one the set O. Now suppose that we create algebras $\mathbb{A}$ and $\mathbb{B}$ by endowing A and B with some dummy operator, say, the identity map. Then we find that the respective ultrafilter frames $\mathbb{A}_\bullet$ and $\mathbb{B}_\bullet$ are isomorphic: both have countably many points, and in both cases, the accessibility relation is simply the diagonal. But the algebras $\mathbb{A}$ and $\mathbb{B}$ are clearly not isomorphic.

As we will see further on, the following theorem from Jónsson & Tarski [70] is not only vital when it comes to applications of the algebraic approach in modal completeness theory. It is also a manifestation of a fundamental mathematical concept, namely that of a *representation* theorem stating that every abstract structure in an axiomatically defined class is in fact isomorphic to a concrete, 'intended' structure of the kind that the axioms try to capture.

THEOREM 54 (Jónsson-Tarski Representation Theorem). *Let $\mathbb{A}$ be a Boolean algebra with τ-operators. Then the Stone representation map $\widehat{\cdot} : A \to \mathcal{P}Uf(\mathbb{A})$ given by*

$$\widehat{a} := \{u \in Uf(\mathbb{A}) \mid a \in u\} \tag{14}$$

is an embedding of $\mathbb{A}$ *into* $(\mathbb{A}_\bullet)^+$.

Proof. We omit details concerning the Boolean part of this theorem, which is of course nothing but Stone's representation theorem for Boolean algebras.

Concerning the additional operations, we restrict ourselves to the basic modal similarity type. So we consider a modal algebra $\mathbb{A} = \langle A, \top, \bot, -, \wedge, \vee, \diamond \rangle$ and show that

$$\widehat{\diamond a} = \langle R_\diamond \rangle \widehat{a}. \tag{15}$$

First we consider an ultrafilter $u \in \langle R_\diamond \rangle \widehat{a}$. It follows by the definition of $\langle R_\diamond \rangle$ (see 9) that there is an ultrafilter v such that $R_\diamond uv$ and $v \in \widehat{a}$, that is, $a \in v$. Then by definition of $R_\diamond$ it follows that $\diamond a \in u$, and, hence, that $u \in \widehat{\diamond a}$. This proves that $\widehat{\diamond a} \subseteq \langle R_\diamond \rangle \widehat{a}$.

For the converse direction, take an arbitrary ultrafilter $u \in \widehat{\diamond a}$; that is, $\diamond a \in u$. We have to come up with an ultrafilter v such that (i) $R_\diamond uv$ and (ii) $v \in \widehat{a}$, or, equivalently, $a \in v$. We first reformulate the first condition:

$$R_\diamond uv \text{ iff } a \in v \text{ for all } a \text{ with } -\diamond - a \in u. \tag{16}$$

Hence, by the Ultrafilter Theorem 50 it suffices to show that the set $\{x \in A \mid -\diamond -x \in u\} \cup \{a\}$ has the finite meet property, see Example 27. In order to prove this, first observe that the set $\{x \in A \mid -\diamond -x \in u\}$ is closed under taking meets — this easily follows from the additivity of $\diamond$ and the fact that u is a filter.

But then it is left to show that $x \wedge a > \bot$ for any $x \in A$ with $-\diamond -x \in u$. Suppose for contradiction that $x \wedge a = \bot$. We obtain $a \leq -x$ so $\diamond a \leq \diamond -x$ by monotonicity of $\diamond$, and so we find $\diamond -x$ in u because $\diamond a \in u$. This gives the desired contradiction since we already had $-\diamond -x$ in u. ❏

DEFINITION 55. Given a Boolean algebra with τ-operators $\mathbb{A}$, the 'double dual' algebra $(\mathbb{A}_\bullet)^+$ is known as the *canonical embedding algebra of* $\mathbb{A}$, the *canonical extension of* $\mathbb{A}$ and the *perfect extension of* $\mathbb{A}$; we will mainly use the second term, and usually denote the structure as $\mathbb{A}^\sigma$.

The Jónsson-Tarski theorem thus states that the constructions $(\cdot)^+$ and $(\cdot)_\bullet$ interact well if we start with algebras: $\mathbb{A} \rightarrowtail (\mathbb{A}_\bullet)^+$ for every BAO $\mathbb{A}$. Unfortunately, if we start with frames, then the return is less safe: for a τ-frame $\mathbb{S}$, the map $s \mapsto \pi_s$ (assigning to points of $\mathbb{S}$ their associated principal ultrafilters) is an embedding of $\mathbb{S}$ into $(\mathbb{S}^+)_\bullet$ only if $\mathbb{S}$ is *image finite*. (In fact, the condition of image-finiteness is also sufficient.) And if $\mathbb{S}$ contains a point from which paths of arbitrary finite length emanate, but no infinite path, then there is *no* bounded morphism from $\mathbb{S}$ to $(\mathbb{S}^+)_\bullet$ whatsoever. From this it follows that there is no way to extend the ultrafilter frame construction to a functor that is *adjoint* to that of taking complex algebras. This is a notable divergence of the case of Boolean algebras per se (that is, without operators) — the formation of the canonical extension $\mathbb{B}^\sigma$ of a Boolean algebra $\mathbb{B}$ is a *free* construction, see [68] for more information on these matters.

Nevertheless, the operation of taking ultrafilter frames can be extended to a functor, as follows.

DEFINITION 56. Let $\mathbb{A}$ and $\mathbb{A}'$ be two Boolean algebras with τ-operators. Given a homomorphism $\eta : \mathbb{A} \to \mathbb{A}'$, we define the map $\eta_\bullet : Uf\,\mathbb{A}' \to \mathcal{P}(A)$, which can be shown to map ultrafilters to ultrafilters, by putting

$$\eta_\bullet(u') := \{a \in A \mid \eta(a) \in u\}. \tag{17}$$

PROPOSITION 57. $(\cdot)_\bullet$ *is a contravariant functor from* BAO_τ *to* Fr_τ.

Proof. If $\eta : \mathbb{A} \to \mathbb{A}'$ is a Boolean homomorphism, then it follows almost immediately that $\eta_\bullet$ maps ultrafilters to ultrafilters, while it is not too hard either to prove that, for any modality ∇ of rank, say, n:

$$\begin{array}{rl} \eta_\bullet \text{ has the forth property for } R_\nabla & \text{if} \quad \nabla'(\eta a_1, \ldots, \eta a_n) \leq \eta(\nabla(a_1, \ldots, a_n)), \\ \eta_\bullet \text{ has the back property for } R_\nabla & \text{if} \quad \nabla'(\eta a_1, \ldots, \eta a_n) \geq \eta(\nabla(a_1, \ldots, a_n)). \end{array}$$

This shows that $\eta_\bullet$ is a bounded morphism from $\mathbb{A}'_\bullet$ to $\mathbb{A}_\bullet$ if $\eta : \mathbb{A} \to \mathbb{A}'$ is a homomorphism. It is then left to show that $(\cdot)_\bullet$ is functorial, and in particular, that $(\eta \circ \theta)_\bullet = \theta_\bullet \circ \eta_\bullet$ for homomorphisms $\theta : \mathbb{A} \to \mathbb{A}'$ and $\eta : \mathbb{A}' \to \mathbb{A}''$. This can be checked by a straightforward calculation which we leave for the reader. ❑

5.4 *Topological duality*

In the previous subsection we encountered a problem of the functor $(\cdot)_\bullet$: in general, algebras cannot be retrieved from their ultrafilter frames. A very simple remedy is then to *add* this information to the frame by melting algebra and frame into one structure. Since this issue already pertains at the level of Boolean algebras (without additional operations), that is where we start the discussion.

DEFINITION 58. A *field of sets* is a pair $\langle S, A\rangle$ with $A \subseteq \mathcal{P}(S)$ being closed under all Boolean set-theoretic operations, or equivalently, with A such that $\langle A, S, \varnothing, \sim_S, \cap, \cup\rangle$ is a subalgebra of $\mathbb{P}S$. The elements of A are called the *admissible* subsets of S.

Given a Boolean algebra $\mathbb{A} = \langle A, \top, \bot, -, \wedge, \vee\rangle$, put $\widehat{A} := \{\widehat{a} \subseteq Uf(\mathbb{A}) \mid a \in A\}$, with $\widehat{\cdot}$ as in (15), and define $\mathbb{A}_* := \langle Uf\mathbb{A}, \widehat{A}\rangle$ as the *associated field of sets* of $\mathbb{A}$. Conversely, the *associated Boolean algebra* of a field of sets $\mathbb{S} = \langle S, A\rangle$ is the structure $\mathbb{S}^* := \langle A, S, \varnothing, \sim_S, \cap, \cup\rangle$.

It will be clear that the Boolean algebras $\mathbb{A}$ and $(\mathbb{A}_*)^*$ will always be isomorphic; however, we will only have that $\mathbb{S} \cong (\mathbb{S}^*)_*$ if $\mathbb{S}$ has some special properties.

DEFINITION 59. A field of sets $\mathbb{S} = \langle S, A\rangle$ is *discrete* if A contains all singletons of S, *differentiated* if for any two distinct points $s \neq t$ of S there is a set $a \in A$ such that $s \in a$ and $t \notin a$, and *full* if $A = \mathcal{P}(S)$. $\mathbb{S}$ is *compact* if every subset of A with the finite intersection property has a non-empty intersection, and *descriptive* if it is both differentiated and compact.

In a descriptive field of sets, the points and the admissible sets are in *balance*: there are sufficiently many admissible sets to separate distinct points, while there are enough points to witness all the ultrafilters of the algebra. More precisely, one can prove that for any field of sets $\mathbb{S} = \langle S, A\rangle$, the map

$$s \mapsto \{a \in A \mid s \in a\} \tag{18}$$

provides an bijection between $\mathbb{S}$ and the collection of ultrafilters of $\mathbb{S}^*$ iff $\mathbb{S}$ is descriptive.

REMARK 60. Our terminology strongly suggests a topological connection. In order to make this explicit, note that the collection of admissible sets of a field of sets $\mathbb{S} = \langle S, A\rangle$ forms a basis for a topology σ_A; and that, conversely, we may take the set $Clp(\mathbb{X})$ of *clopen* (that is, closed and open) elements of a topology $\mathbb{X} = (X, \tau)$ as a collection of

admissible sets. In accordance with this, we define a subset $X \subseteq S$ to be *open* if it is a union of admissible sets, and *closed* if it is an intersection of admissible sets. Thus the study of fields of sets takes us into a rather specific branch of set-theoretic topology in which all spaces are *zero-dimensional*, that is, have a basis of clopens.

One may prove for any field of sets $\mathbb{S} = \langle S, A\rangle$ that $\mathbb{S}$ is descriptive iff $\langle S, \sigma_A\rangle$ is a *Stone space*, that is, σ_A is a compact, Hausdorff and zero-dimensional topology. Basically then, descriptive fields of sets and Stone spaces are two ways of formulating the same mathematical objects; the difference is no more than a matter of focus, be it on the topology itself, or rather on its sets of clopens.

The topological nature also comes out clearly when we discuss *morphisms.*

DEFINITION 61. Given two fields of sets $\mathbb{S} = \langle S, A\rangle$ and $\mathbb{S}' = \langle S', A'\rangle$, we call a map $\theta : S \to S'$ *continuous* if the set

$$\theta^*(a') := \{s \in S \mid \theta(s) \in a'\} \tag{19}$$

belongs to A for all $a' \in A'$.

We define the dual $\eta_* : \mathit{Uf}\,\mathbb{A}' \to \mathit{Uf}\,\mathbb{A}$ of a morphism $\eta : \mathbb{A} \to \mathbb{A}'$ between two Boolean algebras as the map $\eta_*(u') := \{a \in A \mid \eta(a) \in u'\}$.

Without further proof we mention (our reformulation of) the following seminal result from Stone [101] (see Johnstone [68] for an extensive discussion of its impact).

THEOREM 62 (Stone duality). *The functors $(\cdot)^*$ and $(\cdot)_*$ form a dual equivalence between the category of Boolean algebras with homomorphism, and that of descriptive fields of sets with continuous maps.*

The duality for BAOs can now be developed by incorporating the ultrafilter functor $(\cdot)_\bullet$ into the Stone duality: the dual object representing a Boolean algebra with operators will combine the BAO and its dual Kripke frame in one structure.

DEFINITION 63. A *general τ-frame* is a structure $\mathbb{G} = \langle G, R, A\rangle$, where $R = \{R_\nabla \mid \nabla \in \tau\}$ is a family of relations on G, such that (i) $\langle G, R\rangle$ is a τ-frame and (ii) $\langle G, A\rangle$ is a field of sets such that (iii) A is closed under the operation $\langle R_\nabla\rangle$ for each operation symbol $\nabla \in \tau$. The structure $\langle G, R\rangle$ is called the *underlying* Kripke frame of $\mathbb{G}$.

Given a general frame $\mathbb{G} = \langle G, R, A\rangle$, define $\mathbb{G}^*$ as the subalgebra of $\langle G, R\rangle^+$ with carrier A. Conversely, given a τ-BAO $\mathbb{A}$, define its dual general frame $\mathbb{A}_*$ as the structure $\langle \mathit{Uf}(\mathbb{A}), \{R_{\nabla^{\mathbb{A}}} \mid \nabla \in \tau\}, \widehat{A}\rangle$.

As in the case of the duals of Boolean algebras, general frames of the form $\mathbb{A}_*$ are rather special, also with respect to the interaction between their relational and the topological side. We let notions like differentiatedness apply to a general frame $\langle G, R, A\rangle$ as it applies to the underlying field of sets $\langle G, A\rangle$.

DEFINITION 64. A general frame $\mathbb{G} = \langle G, R, A\rangle$ is *tight* if every tuple $(s, s_1, \ldots, s_n)$ which is *not* in the relation R_∇ (with ∇ an arbitrary relation symbol of arity n) is witnessed by admissible sets $a_1, \ldots, a_n$ such that $s_i \in a_i$ for each i, while $s \notin \langle R_\nabla\rangle(a_1, \ldots, a_n)$. $\mathbb{G}$ is *refined* if it is both differentiated and tight, and *descriptive* if it is both refined and compact.

REMARK 65. An easy proof shows that we may reformulate the property of tightness equivalently by requiring that (restricting to the basic modal language here) $R[s] = \bigcap\{a \in A \mid s \in [R]a\}$ for each point s in $\mathbb{G} = \langle G, R, A\rangle$. In other words, the relation R

is *point-closed*, since each point of $\mathbb{G}$ has a *closed* successor set — closed in the induced topology σ_A, that is. Thus from a topological perspective, descriptive general frames can be identified with point-closed relational Stone spaces.

In order to turn the constructions $(\cdot)^*$ and $(\cdot)_*$ into functors we have to introduce morphisms between (descriptive) general frames as well. Again, we combine modal and topological aspects in the natural way.

DEFINITION 66. Given two general frames $\mathbb{G} = \langle G, R, A\rangle$ and $\mathbb{G}' = \langle G', R', A'\rangle$, a map $\theta : G \to G'$ is called a *continuous bounded morphism* if it is both a bounded morphism from $\langle G, R\rangle$ to $\langle G', R'\rangle$ and a continuous map from $\langle G, A\rangle$ to $\langle G', A'\rangle$. The category of descriptive general τ-frames with continuous bounded morphisms is denoted as DGF_τ.

Now let us see how $(\cdot)^*$ and $(\cdot)_*$ operate on morphisms. For the definition of θ^* for θ a continuous bounded morphism we refer to (19); conversely, given a homomorphism $\eta : \mathbb{A} \to \mathbb{A}'$ between two τ-BAOs, define η_* as in Definition 61, that is: $\eta_*(u') := \{a \in A \mid \eta(a) \in u\}$. We have now arrived at the main result of this subsection, Theorem 67 below, which is due to Goldblatt [37, 39]. Independently, Esakia [23] came up with such a duality for a more specific variety of algebras.

THEOREM 67. *The functors $(\cdot)^*$ and $(\cdot)_*$ constitute a dual equivalence between the categories BAO_τ and DGF_τ.*

Proof. It is rather straightforward to verify that $(\cdot)^*$ and $(\cdot)_*$ are functors which form a dual adjunction between the categories DGF_τ and BAO_τ. It is then left to show that $\mathbb{G} \cong (\mathbb{G}^*)_*$ for any descriptive general frame $\mathbb{G}$, and that $\mathbb{A} \cong (\mathbb{A}_*)^*$ for any Boolean algebra with τ-operators $\mathbb{A}$. But both of these claims are easy to establish: for the first isomorphism, take the map of (18); and for the second isomorphism, simply take the Stone embedding $\widehat{\cdot}$ of (14). The proof details are left to the reader. ❑

It is straightforward to derive from this duality that for any class C of general frames, the class of dual algebras algebraizes C (once we have properly defined all notions involved), but we leave the details for the reader.

5.5 Simplicity and Subdirect irreducibility

As an application of these dualities, let us look at the frame counterparts of the notions of simplicity and subdirect irreducibility. In the complex duality of section 5.2, this question has a satisfactory answer, at least for subdirect irreducibility:

THEOREM 68. *Let $\mathbb{S}$ be a τ-frame. Then*

1. *$\mathbb{S}^+$ is simple only if each point is a root of $\mathbb{S}$;*
2. *$\mathbb{S}^+$ is subdirectly irreducible iff $\mathbb{S}$ is rooted.*

Proof. Concerning subdirect irreducibility, the direction from right to left, first mentioned in Goldblatt [39], was already treated in Example 38. The proof of the converse implication appeared first in Sambin [99]. For its details, suppose that ρ is a radix of the algebra $\mathbb{S}^+$, and consider an arbitrary point $s \in S$. Then by definition of radicality we find that $\rho \subseteq \langle R_\blacklozenge\rangle\{s\}$ for some compound modality $\blacklozenge$. It is easy to see that this implies $R_\blacklozenge rs$ for each $r \in \rho$, so that each element of ρ is in fact a root of $\mathbb{S}$. Hence,

if $\mathbb{S}^+$ is simple, then every point is a root of $\mathbb{S}$, since every non-empty subset of S is a radix of $\mathbb{S}^+$ by Theorem 39. If $\mathbb{S}^+$ is s.i., then by the same theorem it has at least one radix; rootedness of $\mathbb{S}$ thus follows from the fact that radical elements are non-empty by definition. ❑

Perhaps contrary to the reader's expectation, the converse of Theorem 68(1) is not true.

EXAMPLE 69. Consider the frame $\mathbb{Z} = (Z, R)$ for the basic modal similarity type, with Z as the set of integers and Rxy iff $|x - y| = 1$. Then clearly every integer is a root of $\mathbb{Z}$, while on the other hand, $\mathbb{Z}^+$ is not simple. An easy way to see this is by proving that the only radical elements of $\mathbb{Z}^+$ are the finite subsets of $\mathbb{Z}$.

In the topological duality of section 5.4, the correspondence between subdirect irreducibility and rootedness is not so nice either. In general, subdirect irreducibility of $\mathbb{A}$ neither implies rootedness of $\mathbb{A}_*$, nor is it implied by it, as the following examples from Sambin [99] witness.

EXAMPLE 70. For an example of the first kind, take the subalgebra $\mathbb{A} \rightarrowtail \langle N, > \rangle^+$ based on the collection of finite and cofinite subsets of the set N of natural numbers. As we will see later on, $\mathbb{A}$ is not subdirectly irreducible. However, the frame $\mathbb{A}_\bullet$ is rooted, since it adds one reflexive point ω (corresponding to the ultrafilter of the cofinite sets) to $\langle N, > \rangle$, in such a way that ω sees all other points.

Conversely, consider the frame $\mathbb{Z}$ of the previous example, and take its subalgebra $\mathbb{B}$ based on the finite and cofinite sets. It is easy to see that $\mathbb{B}$ is s.i.: simply note that every singleton is radical. However, the one reflexive point ∞ that $\mathbb{B}_*$ adds to $\mathbb{Z}$ is not related to any other point in $\mathbb{B}_*$. Hence, $\mathbb{B}_*$ provides an example of an s.i. algebra of which the dual general frame has no roots at all.

These examples indicate that if we are looking for a characterization of the notion of subdirect irreducibility, it does not suffice to look at the dual Kripke frame alone: we have to take the topology into account. Our characterization will be in terms of so-called topological roots or, briefly, topo-roots. Recall that a *root* of a τ-frame $\mathbb{S} = \langle S, R \rangle$ is a point r of $\mathbb{S}$ such that $R^\omega[r] = S$, where the relation R^ω is given as the union of the accessibility relations of the compound diamonds. It is straightforward to verify that in a frame of the form $\mathbb{A}_*$ this boils down to

$$R^\omega uv \text{ iff there is a compound diamond } \blacklozenge \text{ with } \blacklozenge a \in u \text{ for all } a \in v. \tag{20}$$

Our definition of the *topo*-reachability relation is obtained by swapping the universal and the existential quantifier in (20).

DEFINITION 71. Given a Boolean algebra with operators $\mathbb{A}$, define the *topo-reachability* relation $R^\star \subseteq Uf\mathbb{A} \times Uf\mathbb{A}$ as follows:

$$R^\star uv \text{ iff for all } a \in v \text{ there is a compound diamond } \blacklozenge \text{ with } \blacklozenge a \in u. \tag{21}$$

We let $T_{\mathbb{A}}$ denote the set of *topo-roots* of $\mathbb{A}_*$; that is, the collection of those ultrafilters u such that $R^\star[u] = Uf\mathbb{A}$.

The topological terminology will be clarified by the following alternative characterization of $R^\star$.

PROPOSITION 72. *Let $\mathbb{A}$ be some Boolean algebra with τ-operators, and u some ultrafilter of $\mathbb{A}$. Then $R^\star[u] = \overline{R^\omega[u]}$; that is, $R^\star[u]$ is the topological closure of $R^\omega[u]$ in the Stone topology of $\mathbb{A}_*$.*

As the following theorem from Venema [108] witnesses, topo-roots provide the right tool for the characterization of the notions of simplicity and subdirect irreducibility.

THEOREM 73. *Let $\mathbb{A}$ be a Boolean algebra with τ-operators. Then*

1. *$\mathbb{A}$ is simple iff $T_{\mathbb{A}} = Uf\mathbb{A}$;*
2. *$\mathbb{A}$ is subdirectly irreducible iff $T_{\mathbb{A}}$ is open and non-empty.*

Unfortunately, we do not have the space for a proof or even a proof sketch. We confine ourselves to noting that the proof makes use of the correspondence between modal filters of $\mathbb{A}$ and closed, hereditary subsets of $\mathbb{A}_*$.

EXAMPLE 74. It is now obvious why the algebra $\mathbb{A}$ of Example 70 is not s.i.: its dual frame does have a (single) root ω but the set $\{\omega\}$ of roots is not open in the topology of $\mathbb{A}_*$. The algebra $\mathbb{B}$ of the same example on the other hand *is* s.i. Whereas its dual frame $\mathbb{B}_\bullet$ has no *roots* at all, almost every point of $\mathbb{B}_*$ is a topo-root.

As corollaries of the last theorem we obtain some (well-)known results showing that in many cases, nicer characterizations are indeed possible. We call a Boolean algebra with operators *ω-transitive* if it has a *master modality*, that is, a compound diamond $\Diamond$ such that $\blacklozenge a \leq \Diamond a$ for all compound diamonds $\blacklozenge$ and all a in $\mathbb{A}$. (With some authors, this property goes under the name of *weak* transitivity). The following result is due to Sambin [99] (whereas in the closely related field of intuitionistic logic, similar characterizations of s.i. Heyting algebras in terms of their dual structures had been known for some time, cf. Esakia [24]).

COROLLARY 75. *Let $\mathbb{A}$ be an ω-transitive Boolean algebra with operators. Then $\mathbb{A}$ is subdirectly irreducible iff the collection of roots of $\mathbb{A}_*$ is non-empty and open.*

Proof. This follows from Theorem 73 by the observation that if $\mathbb{A}$ is ω-transitive, then $R^\star = R^\omega = R_\Diamond$ (where $\Diamond$ is the master modality of $\mathbb{A}$), whence the notions of root and topo-root coincide. ❑

Results concerning the duals of *finite* BAOs are already covered by Theorem 68, since for finite BAOs the complex and the topological dualities coincide.

5.6 Class operations

While the functors $(\cdot)^+$ and $(\cdot)_\bullet$ do not form a duality, they do provide an interesting link between the categories Fr_τ and BAO_τ. We already discussed the role of the 'double duals', that is, the *canonical embedding algebra* $\mathbb{A}^\sigma = (\mathbb{A}_\bullet)^+$ of a BAO $\mathbb{A}$, and the *ultrafilter extension* $(\mathbb{S}^+)_\bullet$ of a frame $\mathbb{S}$. But there is also a wealth of results concerning the direct interaction of the mentioned functors with the more 'intrinsic' constructions on algebras and frames. We confine ourselves here to the algebraic operations of taking homomorphic images, subalgebras and products, and their frame counterparts of taking generated subframes, bounded morphic images, and disjoint unions. The results listed in

Theorem 76 are more or less direct consequences of the dualities established earlier on; therefore, we leave the proofs to the reader.

THEOREM 76. *Let $\mathbb{S}$, $\mathbb{S}'$ and all $\mathbb{S}_i$ with $i \in I$ be τ-frames, and let $\mathbb{A}$, $\mathbb{A}'$ and all $\mathbb{A}_j$ with $j \in J$ be Boolean algebras with τ-operators. Then*

1. $\theta : \mathbb{S} \rightarrowtail \mathbb{S}'$ *only if* $\theta^+ : \mathbb{S}'^+ \twoheadrightarrow \mathbb{S}^+$;
2. $\theta : \mathbb{S} \twoheadrightarrow \mathbb{S}'$ *only if* $\theta^+ : \mathbb{S}'^+ \rightarrowtail \mathbb{S}^+$;
3. $\eta : \mathbb{A} \rightarrowtail \mathbb{A}'$ *only if* $\eta_\bullet : \mathbb{A}'_\bullet \twoheadrightarrow \mathbb{A}_\bullet$;
4. $\eta : \mathbb{A} \twoheadrightarrow \mathbb{A}'$ *only if* $\eta_\bullet : \mathbb{A}'_\bullet \rightarrowtail \mathbb{A}_\bullet$;
5. $\left(\sum_{i\in I} \mathbb{S}_i\right)^+ \cong \prod_{i\in I} \mathbb{S}_i^+$;
6. $\left(\prod_{j\in J} \mathbb{A}_j\right)_\bullet \cong \sum_{j\in J} (\mathbb{A}_j)_\bullet$

In general it is not true that the ultrafilter frame $(\prod_{j\in J} \mathbb{A}_j)_\bullet$ is isomorphic to the disjoint union $\sum_{j\in J}(\mathbb{A}_j)_\bullet$: the problem is that for infinite J, not every ultrafilter of the product can be linked to an ultrafilter of one of the factors. Fortunately, we do have the following 'second best' connection, essentially due to Gehrke [27], which states that the ultrafilter frame of the product is isomorphic to the disjoint union of the ultrafilter frames of all *ultraproducts* of the original algebras over the index set.

THEOREM 77. *Let $\{\mathbb{A}_i \mid i \in I\}$ be a family of Boolean algebras with τ-operators. Then*

$$\left(\prod_{i\in I} \mathbb{A}_i\right)_\bullet \;\cong\; \sum_{D\in Uf(I)} \left(\prod\nolimits_D \mathbb{A}_i\right)_\bullet .$$

Proof. Given an element a of $\mathbb{A} := \prod_I \mathbb{A}_i$, let $d(a) := \{i \in I \mid a(i) \neq \bot\}$ be the *support set* of a. Then it is not hard to prove that $d[u] := \{d(a) \mid a \in u\}$ is an ultrafilter over I for every $u \in Uf(\mathbb{A})$.

Now given an ultrafilter D over I, the natural homomorphism $\nu^D : a \mapsto a/D$ is a surjective homomorphism from $\mathbb{A}$ onto $\mathbb{A}_D := \prod_D \mathbb{A}_i$. So by Theorem 76(4), its dual $\nu^D_\bullet : (\mathbb{A}_D)_\bullet \to \mathbb{A}_\bullet$ is a frame embedding. We now claim that

$$Range\left(\nu^D_\bullet\right) = \{u \in Uf\mathbb{A} \mid d[u] = D\}. \tag{22}$$

For the inclusion $\subseteq$, take an arbitrary ultrafilter z of $\mathbb{A}_D$. For any $a \in \nu^D_\bullet(z)$, it holds by definition that $\nu^D(a) = a/D$ belongs to z; but then a/D must be distinct from the bottom element of $\mathbb{A}_D$. Hence $d(a) \in D$ by definition of d. Since this applies to arbitrary $a \in \nu^D_\bullet(z)$ it follows that $d[\nu^D_\bullet(z)] \subseteq D$. But then we must have equality because both $d[\nu^D_\bullet(z)]$ and D are ultrafilters over I. For the converse inclusion, if $u \in Uf\mathbb{A}$ satisfies $d[u] = D$, then the set $u_D := \{a/D \mid a \in u\}$ is easily seen to be an ultrafilter of $\mathbb{A}_D$ which satisfies $\nu^D_\bullet(u_D) = u$. This proves (22).

Clearly for each ultrafilter D over I, $Range\left(\nu^D_\bullet\right)$ is (the domain of) a generated subframe of $\mathbb{A}_\bullet$; it now follows from the fact that $d[u] \in Uf(I)$ and (22) that these subframes are mutually disjoint, but jointly cover the full domain $Uf\mathbb{A}$ of $\mathbb{A}_\bullet$. From this the theorem is immediate. ❑

On the basis of the Theorems 76 and 77 we may develop a 'calculus of class operations'. For instance, letting $\mathsf{S_f}$ denote the operation of taking generated subframes, Theorem 76(1) can be read as stating '$\mathsf{CmS_f} \leq \mathsf{HCm}$', meaning that $\mathsf{CmS_f}(\mathsf{C}) \subseteq \mathsf{HCm}(\mathsf{C})$ for every frame class C. There are many constructions of either frames or algebras that have been investigated, and many results, similar to the Theorems 76 and 77, have been obtained. The interested reader is referred to work by Goldblatt, for instance [40, 41].

Unfortunately, we have only space here for one further example (which will be used in the next section).

PROPOSITION 78. *For any class* C *of frames,* $\mathsf{PuCm}(\mathsf{C}) \subseteq \mathsf{SCmPu}(\mathsf{C})$.

Proof. Let $\{\mathbb{S}_i \mid i \in I\}$ be a family of τ-frames, and let D be an ultrafilter over I. Define the map $\eta : \prod_I \mathcal{P}(S_i)/D \to \mathcal{P}(\prod_I \mathbb{S}_i/D)$ by putting, for s/D in $\prod_I \mathbb{S}_i/D$:

$$s/D \in \eta(a/D) \;:\Longleftrightarrow\; \{i \in I \mid s(i) \in a(i)\} \in D.$$

We leave it for the reader to verify that this is a well-defined embedding of $\prod_I \mathbb{S}_i^+/D$ into $(\prod_I \mathbb{S}_i/D)^+$. ❑

We will give one application of the Theorems 76 and 77 here, more use of these results will be made in the next sections. Theorem 79 below, due to Goldblatt & Thomason [47], can be read as a modal dual of Birkhoff's theorem identifying varieties with equational classes. For a definition of Birkhoff's theorem from a *coalgebraic* perspective, the reader is referred to section 14.

THEOREM 79 (Goldblatt-Thomason Theorem). *Let* C *be a class of* τ*-frames. Then*

1. *if* C *is modally definable then it reflects ultrafilter extensions, and is closed under taking bounded morphic images, generated subframes and disjoint unions;*

2. *the converse of (1) holds if* C *is closed under taking ultrapowers, (for instance, if* C *is elementary).*

Proof. First assume that C is modally definable; that is, $\mathsf{C} = \mathsf{Fr}(\Gamma)$ for some set Γ of modal τ-formulas (in fact, we may take Γ to the logic of C, but this is not relevant now). Now suppose that the frame $\mathbb{S}'$ is the bounded morphic image of some $\mathbb{S}$ in C. From $\mathbb{S}$ in C it follows that $\mathbb{S} \Vdash \Gamma$ whence $\mathbb{S}^+ \models \Gamma^{\approx}$; but at the same time we see that by Theorem 76(2), $\mathbb{S}'^+$ is a subalgebra of $\mathbb{S}^+$. Hence also $\mathbb{S}'^+ \models \Gamma^{\approx}$, so $\mathbb{S}' \models \Gamma$ which immediately implies that $\mathbb{S}'$ belongs to C. This shows that C is closed under taking bounded morphic images; the case of generated subframes and disjoint unions is proved similarly.

Now suppose that the ultrafilter extension $\mathbf{ue}\mathbb{S} = (\mathbb{S}^+)_\bullet$ belongs to C. Then $((\mathbb{S}^+)_\bullet)^+ \models \Gamma^{\approx}$, and so $\mathbb{S}^+ \Vdash \Gamma^{\approx}$ since $\mathbb{S}^+$ is a subalgebra of $((\mathbb{S}^+)_\bullet)^+$ by the Jónsson-Tarski Theorem 54. But from $\mathbb{S}^+ \models \Gamma^{\approx}$ it follows that $\mathbb{S} \Vdash \Gamma$ whence $\mathbb{S}$ belongs to C. This shows that C reflects ultrafilter extensions, and thus proves part (1).

For the second part, assume that C enjoys all of the listed closure properties. In order to prove that $\mathsf{C} = \mathsf{Fr}(\mathit{Log}(\mathsf{C}))$, take an arbitrary frame $\mathbb{S}$ such that $\mathbb{S} \Vdash \mathit{Log}(\mathsf{C})$. It suffices to show that $\mathbb{S}$ actually belongs to C.

It follows from $\mathbb{S} \Vdash \mathit{Log}(\mathsf{C})$ that $\mathbb{S}^+$ validates the equational theory of the class $\mathsf{Cm}(\mathsf{C})$, and so by Birkhoff's variety theorem $\mathbb{S}^+$ belongs to the variety $\mathsf{VarCm}(\mathsf{C})$ generated by

the class of complex algebras over C. Then by Tarski's HSP-theorem, $\mathbb{S}^+$ belongs to $\mathsf{HSPCm}(\mathsf{C})$. That is, for some family $\{\mathbb{F}_i \mid i \in I\}$ of frames in C, and some algebra $\mathbb{A}$ we have that

$$\mathbb{S}^+ \twoheadleftarrow \mathbb{A} \rightarrowtail \prod\nolimits_I \mathbb{F}_i^+.$$

Note that $\prod_I \mathbb{F}_i^+ \cong (\sum_I \mathbb{F}_i)^+$ by Theorem 76(5), and that $\mathbb{F} := \sum_I \mathbb{F}_i$ belongs to C. Then using Theorem 76(3) and (4) we find that

$$(\mathbb{S}^+)_\bullet \rightarrowtail \mathbb{A}_\bullet \twoheadleftarrow (\mathbb{F}^+)_\bullet.$$

Now it follows by Theorem 90 in Chapter 5 of this volume that $(\mathbb{F}^+)_\bullet$ is a bounded morphic image of some *ultrapower* $\mathbb{F}^J/D$ of $\mathbb{F}$. Then by the various listed closure properties of C, we show that subsequently, each of the frames $\mathbb{F}^J/D$, $(\mathbb{F}^+)_\bullet$, $\mathbb{A}_\bullet$ and $(\mathbb{S}^+)_\bullet$ belong to C. Finally then, also $\mathbb{S}$ belongs to C since its ultrafilter extension $(\mathbb{S}^+)_\bullet$ does so. ❑

6 LOGICS AND VARIETIES

This section, which forms the heart of the algebra part of this chapter, discusses the connection between normal modal logics (NMLs) and varieties of BAOs. The main part of the section consists in showing how standard properties of a logic turn up on the algebraic side of the picture, but we start with showing how the *lattice* of normal modal logics is dually isomorphic to that of the varieties of BAOs.

DEFINITION 80. Given a normal modal logic L, we say that a normal modal logic L' is a *normal extension* of L simply if $L \subseteq L'$. The lattice of normal extensions of L is denoted as $\mathrm{NExt}(L)$.

We have already seen that with every normal modal τ-logic we may associate a variety $\mathsf{BAO}_\tau(L)$ of τ-BAOs. Conversely, every class of these algebras gives rise to a normal modal logic.

DEFINITION 81. Given a class K of Boolean algebras with τ-operators, we define $Log(\mathsf{K}) := \{\varphi \in Fma_\tau \mid \mathsf{K} \models \varphi^\approx\}$.

The following theorem then describes the intimate connection between normal modal logics and varieties of BAOs. Similar results can be proved about arbitrary modal logics and varieties of BAEs, and about monotone modal logics and varieties of BAMs.

THEOREM 82.

1. *The maps $\mathsf{BAO}_\tau(\cdot)$ and $Log(\cdot)$ form a Galois connection, in the sense that for every set Γ of τ-formulas, and every class K of Boolean algebras with τ-operators, $\Gamma \subseteq Log(\mathsf{K})$ iff $\mathsf{K} \subseteq \mathsf{BAO}_\tau(\Gamma)$.*

2. *The stable formula sets of this connection are precisely the normal modal τ-logics, while the stable classes of algebras are precisely the varieties of Boolean algebras with τ-operators.*

3. *Hence, Log is a dual isomorphism between the lattice of subvarieties of BAO_τ and the lattice $\mathrm{NExt}(\mathbf{K}_\tau)$ of normal modal τ-logics*

Proof. It is not hard to see the Galois connection, since we have $\Gamma \subseteq \mathit{Log}(\mathsf{K})$ iff $\mathbb{A} \models \gamma^{\approx}$ for all $\mathbb{A}$ in K and all $\gamma \in \Gamma$ iff $\mathsf{K} \subseteq \mathsf{BAO}_\tau(\Gamma)$.

Now let Γ be a stable set of formulas of this connection, that is, suppose that $\Gamma = \mathit{Log}(\mathsf{BAO}_\tau(\Gamma))$; one easily infers that such a Γ must be a normal modal logic. Conversely, if L is a normal modal logic, then $L = \mathit{Log}(\mathsf{BAO}_\tau(L))$ by the Algebraization Theorem 17.

At the other side of the connection, it is immediate from the definition that every class $\mathsf{BAO}_\tau(\Gamma)$ is a variety. Conversely, assume that V is a variety of τ-BAOs. Then clearly $\mathsf{V} \subseteq \mathsf{BAO}_\tau(\mathit{Log}(\mathsf{V}))$ since this holds for any class; for the opposite inclusion, by Birkhoff's variety theorem it suffices to show that $\mathsf{BAO}_\tau(\mathit{Log}(\mathsf{V}))$ validates every equation of V. So suppose that $\mathsf{V} \models \varphi \approx \psi$; then $\mathsf{V} \models (\varphi \leftrightarrow \psi) \approx \top$ since V has a Boolean basis; from this it follows that $\varphi \leftrightarrow \psi \in \mathit{Log}(\mathsf{V})$, whence $\mathsf{BAO}_\tau(\mathit{Log}(\mathsf{V}))$ validates the equation $(\varphi \leftrightarrow \psi) \approx \top$, by definition. But $\mathsf{BAO}_\tau(\mathit{Log}(\mathsf{V}))$ also has a Boolean basis, so we find that $\mathsf{BAO}_\tau(\mathit{Log}(\mathsf{V})) \models \varphi \approx \psi$, as required.

The last part of the theorem is then immediate by the general theory of Galois connections. ❑

The dual isomorphism given by Theorem 82, linking the lattice of normal modal logics to that of varieties of BAOs, has yielded a wealth of information on modal logics. For instance, universal algebraic theory on *splitting algebras* led algebraically minded modal logicians to strong results on the *degree of Kripke incompleteness* of a modal logic, see for instance Blok [15]. We will not discuss the lattice of modal logics any further in this chapter, referring the reader to the Chapters 7 and 8 of this volume.

Instead we turn to the question, how standard logical phenomena fit in the algebraic framework presented so far. The answer to this question depends on the issue at stake, so let us consider a number of examples:

completeness is a property not so much of a single logic but rather of a pair of logics. For instance, Kripke completeness of a logic L means that L coincides with the logic of its frame class C. Algebraically, this corresponds to the fact that the variety $\mathsf{BAO}_\tau(L)$ is generated by the class of complex algebras $\mathsf{Cm}(\mathsf{C})$. More details will be provided in subsection 6.1.

canonicity of a modal logic L has, as we will see in subsection 6.2, an algebraic counterpart in the property of a class of algebras being closed under taking canonical extensions.

correspondence is more about formulas, or equations, than about logics, or varieties of algebras. Nevertheless, it has a clear algebraic meaning: We can say that an equation $s \approx t$ *corresponds, over a frame class* C to a first-order formula α in the language of frames, if, for all frames $\mathbb{S}$ in C, we have that $\mathbb{S}^+ \models s \approx t$ iff $\mathbb{S} \models \alpha$.

interpolation is a property of a normal modal logic. In subsection 6.3 we will see that it corresponds to an amalgamation property on the algebraic side.

Let us now move to a more detailed discussion of some of these issues.

6.1 Completeness

As we mentioned already, Theorem 17 can be read as a *general* algebraic *completeness* result. So in this respect the algebraic semantics behaves much better than the relational

one: Classes of Kripke frames are generally not adequate for revealing all distinctions between normal modal logics, see Chapter 7 of this volume for the details. It clearly *means* something for a modal logic to be Kripke complete, so what about the associated algebraic variety? For an answer, recall the notion of a *perfect* BAO from Definition 40.

THEOREM 83. *A normal modal τ-logic L is (Kripke) complete iff* $\mathsf{BAO}_\tau(L)$ *is generated by its perfect members.*

Proof. Straightforward by the observation that any variety V of BAOs is generated by its perfect members iff its equational theory coincides with that of the class $\mathsf{CmStr}(\mathsf{V})$. ❑

This inspires the following definition.

DEFINITION 84. A variety V of Boolean algebras with τ-operators is called *(Kripke) complete* if V is generated by its perfect members.

The phenomenon of Kripke incompleteness of normal modal logics is thus algebraically reflected by the fact that many different varieties of BAOs may share the same class of perfect members.

The formulation of Theorem 83 strongly suggests that Kripke completeness is only one of a *family* of properties pertaining to normal modal logics. In fact, one may wonder whether varieties of Boolean algebras with operators are generated by those of their members that meet any given constraint. For instance, we might consider varieties that are generated by their *finite* members. Since every finite BAO is perfect this gives a strong version of Kripke completeness that is known on the logical side as the *finite model property* of the logic.

In this respect it is also interesting to see what happens if we consider *weakenings* or *variations* of the notion of perfection. For instance, recall that perfection of a BAO is the conjunction of three properties: atomicity and completeness of the underlying Boolean algebra, and complete additivity of the operators. Hence, we may naturally ask which varieties of BAOs are generated by their atomic members, their complete and completely additive members, etc. Recent investigations have provided answers to some of these questions. First however, we mention a result of Buszkowski [18] which has been around for almost twenty years already, but which seems to have received little attention. Call a first-order formula or equation in the language of Boolean algebras with operators *modally guarded* if every variable occurs within the scope of a modality.

THEOREM 85. *Let* V *be a variety of expanded Boolean algebras which is axiomatized by modally guarded equations. Then* V *is generated by its atomic members.*

Proof. Given two BAEs $\mathbb{A}$ and $\mathbb{A}'$, call an embedding $\eta : \mathbb{A} \rightarrowtail \mathbb{A}'$ *guarded* if for all guarded formulas $\varphi(x_1, \ldots, x_k)$, and all $a_1, \ldots, a_k \in A$, it holds that $\mathbb{A} \models \varphi[a_1, \ldots, a_k]$ iff $\mathbb{A}' \models \varphi[\eta a_1, \ldots, \eta a_k]$. Then

$$\text{every BAE } \mathbb{A} \text{ has a guarded embedding into an atomic BAE.} \tag{23}$$

It is straightforward to prove the theorem from (23): Any algebra $\mathbb{A}$ in V can be embedded into an atomic BAE $\mathbb{B}$ that satisfies the same guarded sentences as $\mathbb{A}$, and thus in particular, also belongs to V.

For a proof of (23), let $\mathbb{A}$ be some τ-expanded Boolean algebra. By the Stone representation theorem, we may assume that for some set X, $\mathbb{A}$ is of the form

$$\langle A, X, \varnothing, \sim_X, \cup, \cap, \{\nabla^{\mathbb{A}} \mid \nabla \in \tau\}\rangle.$$

In fact, we may assume that every non-empty $a \in A$ is an *infinite* subset of X. (Otherwise, replace X with the set $X \times \omega$ and, using the natural embedding $P \mapsto P \times \omega$ of the power set algebra of X into that of $X \times \omega$, continue with the image of $\mathbb{A}$ under this map.) Now let B be the collection of those subsets b of X that differ in at most finitely many elements from some element of A; that is,

$$B := \{b \subseteq X \mid (a \cap \sim_X b) \cup (b \cap \sim_X a) \text{ is finite, for some } a \in A\}.$$

It is not hard to see that for every $b \in B$ there is in fact a *unique* element $a \in A$ such that the symmetric difference $(a \cap \sim_X b) \cup (b \cap \sim_X a)$ is finite; this element will be denoted as b^*.

One then easily proves that the structure $\langle B, X, \varnothing, \sim_X, \cup, \cap\rangle$ is an atomic Boolean algebra, so if we define, for $\nabla \in \tau$:

$$\nabla^{\mathbb{B}}(b_1, \ldots, b_n) := \nabla^{\mathbb{A}}(b_1^*, \ldots, b_n^*),$$

we obtain a τ-expanded Boolean algebra $\mathbb{B}$. Finally, a straightforward induction on the complexity of guarded formulas shows that the identity map is the required guarded embedding of $\mathbb{A}$ into $\mathbb{B}$. This proves (23). ❑

However, the restriction to *guarded* axioms in Theorem 85 is essential, as the following result of Venema [106] implies that there are varieties of BAOs that have *no* atomic members.

THEOREM 86. *There are nontrivial varieties of Boolean algebras with operators of which all members are atomless.*

Proof. The basic idea underlying this proof is straightforward: construct a particular, nontrivial, BAO $\mathbb{A}$, and a unary term $\pi(x)$ such that the formula $\alpha \equiv \forall x(\bot \prec x \Rightarrow \bot \prec \pi(x) \prec x)$ holds in $\mathbb{A}$. This shows not only that $\mathbb{A}$ is atomless, but that this atomlessness is witnessed by a *term function.*

Lacking the space for further details concerning the construction of $\mathbb{A}$, we briefly sketch how to prove the theorem from here. Let K be the class of BAOs satisfying α. Without loss of generality, assume that K has a global modality (see section 8.2). It then follows that the class $\mathsf{SP}(\mathsf{K})$ is a variety, and thus, that the formula α, being a universal Horn sentence, holds in every member of this variety. But then every such algebra is atomless, so the theorem follows if we can prove that K is nontrivial. But this is an immediate consequence of the existence of the algebra $\mathbb{A}$. ❑

Regarding the order/lattice theoretic property of completeness, a similar result obtains, due to Litak [81].

THEOREM 87. *There are nontrivial varieties of Boolean algebras with operators without complete members.*

Proof. Consider the similarity type of *tense logic*, as in section 8.1. Let $\mathbb{S} = \langle N, < \rangle$ be the bidirectional frame of the natural numbers with the standard ordering. That is, we interpret the diamonds $\Diamond_F$ and $\Diamond_P$ via the relations $<$ and $>$, respectively. Furthermore, let $\mathbb{A}$ be the subalgebra of $\mathbb{S}^+$ based on the collection of finite and cofinite subsets of $\mathbb{N}$. We claim that $\mathsf{Var}(\mathbb{A})$, the variety generated by $\mathbb{A}$, has no complete members. Suppose for contradiction that $\mathbb{C}$ is a complete member of $\mathsf{Var}(\mathbb{A})$.

Each natural number n is, inside $\mathbb{S}$, the unique point satisfying the variable free formula $\varphi_n := \Diamond_P^n \top \wedge \Box_P^{n+1} \bot$. Observe that the inequalities $\varphi_n \wedge \varphi_m \preceq \bot$ (for $m \neq n$), and $\varphi_n \preceq \Diamond_F \varphi_{n+1}$ hold in $\mathbb{A}$, hence in $\mathsf{Var}(\mathbb{A})$, and therefore, in $\mathbb{C}$. Define $a_n := \varphi_{2n}^{\mathbb{C}}$ and $b_n := \varphi_{2n+1}^{\mathbb{C}}$. It is then immediate that $a_n \leq \Diamond_F b_n$, $b_n \leq \Diamond_F a_{n+1}$, and $a_n \wedge b_m = \bot$, for all m, n (we write $\Diamond_F$ rather than $\Diamond_F^{\mathbb{C}}$). But $\mathbb{C}$ is complete, so it contains elements $a = \bigvee_n a_n$ and $b = \bigvee_n b_n$, for which we easily derive that $a \leq \Diamond_F b$, $b \leq \Diamond_F a$, and $a \wedge b = \bot$. Hence, from the fact that $\mathbb{C} \models \Diamond_F \Diamond_F x \preceq \Diamond_F x$ it follows that $a \leq \Diamond_F a \wedge \Diamond_F -a$, whence $a \wedge (\Box_F a \vee \Box_F -a) = \bot$. Thus $\mathbb{C}$ refutes the inequality $\Diamond_F x \preceq \Diamond_F (x \wedge \Box_F x \vee \Box_F \neg x)$, while a straightforward proof shows this inequality to hold in $\mathbb{A}$, and hence, in $\mathsf{Var}(\mathbb{A})$. This provides the required contradiction. ❑

For more information on such notions of incompleteness that are weaker than Kripke incompleteness, the reader is referred to Litak [81]. To mention one open problem: it is not known whether an analogue of the previous two results can be proved for the notion of completely additivity.

6.2 Canonicity

In Chapter 2 of this volume, a normal modal logic L is defined to be *canonical* if $\mathbb{C}_L \Vdash L$, where $\mathbb{C}_L$ is the canonical frame for the logic L. In order to put this in an algebraic perspective, first note that $\mathbb{C}_L \Vdash L$ is equivalent to the requirement that $\mathbb{C}_L^+ \models L^{\approx}$. Also, recall from Example 52 that the canonical frame for L is isomorphic to the ultrafilter frame of the Lindenbaum-Tarski algebra $\mathbb{F}_L$. Hence, we see that the issue is whether $(\mathbb{F}_L)^\sigma = ((\mathbb{F}_L)_\bullet)^+ \models L^{\approx}$, whereas we know that $\mathbb{F}_L \models L^{\approx}$, cf. Theorem 21. This inspires the following definition.

DEFINITION 88. A class of Boolean algebras with τ-operators is *canonical* if it is closed under taking canonical embedding algebras. Accordingly, an equation η is called *canonical* if the variety $\mathsf{BAO}_\tau(\eta)$ is canonical, that is, if $\mathbb{A} \models \eta$ only if $\mathbb{A}^\sigma \models \eta$, for all BAOs $\mathbb{A}$.

From the definition it is obvious that any normal modal logic is canonical if the variety $\mathsf{BAO}_\tau(L)$ is canonical, but what about the converse implication? Here we need to be a bit more precise about the definition of the canonical frame; in particular, about the size of the set of variables. For, observe that the notion of *maximality* of an L-consistent set of formulas depends on the surrounding set of formulas, and hence, on the set X of variables. Thus the shape of the canonical frame $\mathbb{C}_L$ depends on the *size* of the set X of variables; in order to make this dependence explicit, we will write $\mathbb{C}_L(X)$ for the canonical frame in which the points are maximal L-consistent subsets of $Fma(X)$. A similarly convention applies to Lindenbaum-Tarski algebras. Taking this cardinal subtlety into account, we arrive at a sharpened definition of the logical concept of canonicity.

DEFINITION 89. A normal modal logic L is *canonical* if $\mathbb{C}_L(X) \Vdash L$ for *all* sets X. A formula φ is called canonical if $\mathbb{C}_L(X) \Vdash \varphi$ for all normal modal logics L containing φ.

Fortunately, we can prove that the logical and the algebraic notion of canonicity coincide.

THEOREM 90. *For any normal modal τ-logic L, L is canonical iff* $\mathsf{BAO}_\tau(L)$ *is a canonical variety.*

Proof. Let $\mathbb{A}$ be an arbitrary algebra in $\mathsf{BAO}_\tau(L)$, and let X be a set containing a separate variable x_a for each $a \in A$. Then $\mathbb{A}$ is a homomorphic image of $\mathbb{F}_L(X)$ by the fact that $\mathbb{F}_L(X)$ is the free algebra for $\mathsf{BAO}_\tau(L)$ over the set $[X]_L$, see Theorem 22 for the case of countable X. Now two applications of Theorem 76 show that $(\mathbb{F}_L(X))^\sigma \twoheadrightarrow \mathbb{A}^\sigma$. But $(\mathbb{F}_L(X))^\sigma$ belongs to $\mathsf{BAO}_\tau(L)$ by canonicity of L, and so $\mathbb{A}^\sigma$ is in $\mathsf{BAO}_\tau(L)$ because varieties are closed under taking homomorphic images. ❑

It is not known whether, for the variety $\mathsf{BAO}_\tau(L)$ to be canonical, it suffices that the canonical frames for countable variable sets validate L. Leaving this question as an open problem, we turn to the logical *motivation* of the concept of canonicity. This lies in its applications in modal completeness theory, see Chapter 2 of this volume for details. Algebraically, these applications are connected to the following result.

THEOREM 91. *Let* V *be a variety of Boolean algebras with τ-operators. If* V *is canonical, then* V *is complete.*

Proof. If V is canonical then $\mathsf{V} \subseteq \mathsf{SCmCst}(\mathsf{V})$ so clearly V is generated by its perfect members. ❑

So where do we find canonical varieties? In general there seem to be two roads here, a syntactic and a model-theoretic one. The syntactic approach is the most important one for applications. Basically, the idea is to find out whether a logic is canonical on the basis of the syntactic shape of the axioms. Now in general it is *undecidable* whether a given formula φ is canonical (see Kracht [72] for a proof). Fortunately, however, there are fairly large classes of canonical formulas that occur frequently in practice, and are easily recognized. We confine our attention here to *Sahlqvist formulas* — these are also discussed in the Chapters 1, 5 and 7 of this volume.

In the sequel it will be convenient to assume that the primitive symbols of our language are, besides the Boolean connectives $\top$, $\bot$, $\neg$, $\wedge$ and $\vee$, and the modalities $\{\nabla \mid \nabla \in \tau\}$, also the implication symbol $\to$, and the dual modalities $\{\Delta \mid \nabla \in \tau\}$. Also, recall that boxes are the duals of diamonds, that is, of *unary* modal operators.

DEFINITION 92. Given a modal similarity type τ, we define the following classes of terms/formulas. A *boxed atom* is a variable, possibly preceded by a string of boxes. A formula π is *positive* (*negative*) if all of its variables are in the scope of an even (odd, respectively) number of negation symbols. A *Sahlqvist formula* is a formula of the form $\varphi \to \psi$, where φ is built up from negative formulas, boxed atoms, and constants, using only modalities, $\wedge$ and $\vee$, while ψ is a positive formula.

The following results are some of the most celebrated general results in modal logic. Theorem 93 below, from Sahlqvist [98], put the crown on the work of many contemporary modal logicians.

THEOREM 93 (Sahlqvist Canonicity). *Every Sahlqvist formula is canonical.*

For the *proof*, the reader is referred to section 7. As a corollary of this theorem and the *correspondence* result for Sahlqvist formulas (see Chapter 1 of this volume), we obtain the following.

COROLLARY 94. *Let $L = \mathbf{K}_\tau.\Sigma$ be a normal modal logic axiomatized by a collection Σ of Sahlqvist axioms. Then L is sound and complete with respect to the class of frames defined by the first-order correspondents of the formulas of Σ.*

REMARK 95. Although the Sahlqvist canonicity theorem takes care of most of the canonical formulas that one encounters in practice, it certainly does not cover the concept completely. For instance, Goranko & Vakarelov [49] widen the class to that of so-called *inductive* formulas, see Chapter 5 of this volume for some discussion. Jónsson [69] generalizes an example of Fine [25] to the result that for every positive formula $\varphi(x)$, the equation $\varphi(x \vee y) \approx \varphi(x) \vee \varphi(y)$ is canonical. And of course, there are individual examples of canonical formulas, such as the conjunction of the transitivity axiom 4 and the McKinsey axiom $\Box\Diamond x \leq \Diamond\Box x$, cf. [69] for an algebraic proof.

As we mentioned, a second way to arrive at canonical varieties of BAOs proceeds via a model-theoretic road. The basic idea here is that varieties are canonical if they can be *generated* in a certain way. A first and seminal result in this direction was the following.

THEOREM 96 (Fine). *If K is an elementary class of frames, then $Log(\mathsf{K})$ is a canonical normal modal logic.*

Algebraically, Theorem 96 reads that elementary frame classes generate canonical varieties. This result points at an intriguing connection between elementary frame classes and canonical varieties. In particular, it has been an open problem for a long time whether the converse of Fine's theorem would hold as well, that is, whether every canonical variety would be generated by some elementary frame class. Recently however, this issue has been settled negatively in Goldblatt, Hodkinson & Venema [46].

THEOREM 97. *There is a canonical variety that is not generated by any elementary frame class.*

Proof. The example that we give here is based on a famous graph-theoretic result due to Erdős. Here a *graph* is a pair $\mathbb{G} = (G, E)$ with E an irreflexive, symmetric relation on G. A *k-coloring* of $\mathbb{G}$ is a partition of G into k *independent* sets, i.e., sets containing no pair of neighboring vertices. The *chromatic number* $\chi(\mathbb{G})$ of $\mathbb{G}$ is the smallest number k for which it has a k-coloring, and ∞ if it has no finite coloring. A *cycle* in $\mathbb{G}$ is a path $x_1 E x_2 E \ldots E x_n E x_1$ such that $n \geq 3$ and $x_1, \ldots, x_n$ are all distinct vertices; the length of this cycle is n.

Now intuitively, a lack of short cycles, indicating a certain 'looseness' of the graph, should make it easy to color a graph with few colors, but Erdős [22] reveals the existence of a sequence of finite graphs whose n-th member $\mathbb{G}_n$ has chromaticity bigger than n while $\mathbb{G}_n$ has *no* cycles of length $\leq n$. Fix such a sequence $\{\mathbb{G}_n \mid n \geq 2\}$, under the additional assumption that $|\mathbb{G}_n| > |\mathbb{G}_m|$ if $n > m$. (Here $|\mathbb{G}|$ denotes the number of vertices in $\mathbb{G}$.)

The modal similarity type ϵ of our variety EG will have two diamonds, $\Diamond$ and E. On a graph $\mathbb{G}$, the first of these will be interpreted through the edge relation, and the second, through the *global relation* $\Upsilon_G = G \times G$. That is, E is a *global modality*, cf. section 8.2. In the sequel we will blur the distinction between the structures $\langle G, E, \Upsilon_G \rangle$ and $\langle G, E \rangle$,

for instance calling $\langle G, E, \Upsilon_G\rangle^+$ the complex algebra of $\mathbb{G}$, and denoting it, accordingly, as $\mathbb{G}^+$.

For the definition of EG we extend the notion of chromaticity to arbitrary algebras. An element a of an ϵ-BAO $\mathbb{A}$ is called *independent* if $a \wedge \Diamond a = \bot$; write $\chi(\mathbb{A})$ for the chromatic number of $\mathbb{A}$, that is, for the least k such that there are independent $a_1, \ldots, a_k$ with $a_1 \vee \cdots \vee a_k = \top$ and $a_i \wedge a_j = \bot$ for $i \neq j$, putting $\chi(\mathbb{A}) = \infty$ if there is no finite such k. Note that this definition generalizes the one given earlier, in the sense that for any graph $\mathbb{G}$, $\chi(\mathbb{G}) = \chi(\mathbb{G}^+)$.

Now let $\psi_{n,m}$ be the first order formula in this algebraic language stating that if $\mathbb{A}$ has at least 2^n elements, then $\chi(\mathbb{A}) > m$, and define

$$\begin{array}{rcl}\Psi & := & \{\psi_{1,2}\} \cup \{\psi_{|\mathbb{G}_n|,n} \mid n \geq 2\}, \\ \Gamma & := & \{x \preceq \mathsf{E}x, \mathsf{E}\mathsf{E}x \preceq \mathsf{E}x, \mathsf{E}\neg\mathsf{E}\neg x \preceq x, \Diamond x \preceq \mathsf{E}x\}.\end{array}$$

Note that Γ is the set of equations defining E to be a global modality, cf. Definition 135 for the logical incarnation of Γ. Let C denote the class of algebras satisfying the formulas $\Psi \cup \Gamma$, and let EG denote the variety generated by C. It follows from Theorem 139 that $\mathsf{EG} = \mathsf{SP}(\mathsf{C})$.

We first show that EG is canonical. Note that since C is an elementary class, it suffices by Theorem 98 below to prove that C itself is canonical. Take an arbitrary algebra $\mathbb{A}$ in C. If $\mathbb{A}$ is finite, then $\mathbb{A}^\sigma \cong \mathbb{A}$ is in C by assumption. If $\mathbb{A}$ is infinite, then $|\mathbb{A}| > 2^{|\mathbb{G}_n|}$ for all $n \geq 2$, so by $\mathbb{A} \models \psi_{|\mathbb{G}_n|,n}$ we obtain that $\chi(\mathbb{A}) > n$ for all $n \geq 2$. Clearly then $\chi(\mathbb{A}) = \infty$; from this we may derive that the ultrafilter frame $\mathbb{A}_\bullet$ has a *reflexive* point, which implies that $(\mathbb{A}_\bullet)^+$, being the complex algebra of $\mathbb{A}_\bullet$, has infinite chromaticity as well. But then we see that $\mathbb{A}^\sigma \models \psi_{m,n}$ for all m, n, so we certainly have $\mathbb{A}^\sigma \models \Psi$. It is easily seen that the formulas Γ are canonical, so that we have proved that $\mathbb{A}^\sigma$ belongs to C.

It is left to prove that EG is not elementarily generated. Theorem 4.12 of Goldblatt [40] states that any variety V of BAOs which is elementarily generated, is generated by an elementary frame class K such that $\mathsf{Cst}(\mathsf{V}) \subseteq \mathsf{K} \subseteq \mathsf{Str}(\mathsf{V})$. Hence, for our purpose it suffices to come up with a family of frames in $\mathsf{Cst}(\mathsf{EG})$ that provide an ultraproduct outside $\mathsf{Str}(\mathsf{EG})$, and the obvious candidates for this are the Erdős frames $\{\mathbb{G}_n \mid n \geq 2\}$. It is easy to check that $\mathbb{G}_n^+ \models \Psi$ for each $n \geq 2$, so each $\mathbb{G}_n^+$ belongs to C. But then all Erdős frames belong to $\mathsf{Cst}(\mathsf{C})$, because each $\mathbb{G}_n$, being finite, is isomorphic to $(\mathbb{G}_n^+)_\bullet$. Now take a non-principal ultrafilter D over the set $\omega \setminus \{0,1\}$. Observe that for each k, only finitely many of the $\mathbb{G}_n$ have any cycles of length k; hence, by Łos' theorem, the ultraproduct $\prod_D \mathbb{G}_n$ has no cycles at all, and hence, it is *2-colorable.*

This shows that $\prod_D \mathbb{G}_n$ does not belong to C, since it follows from $\mathsf{C} \models \psi_{1,2}$ that every nontrivial algebra in C has chromaticity at least three. But fairly direct proofs show that $\chi(\prod_I \mathbb{A}_i) \geq \chi(\mathbb{A}_i)$ for all i, and that $\chi(\mathbb{A}) \geq \chi(\mathbb{A}')$ if $\mathbb{A} \rightarrowtail \mathbb{A}'$. This implies that $\chi(\mathbb{A}) > 2$ for all $\mathbb{A}$ in $\mathsf{SP}(\mathsf{C})$, so by the fact that $\mathsf{SP}(\mathsf{C}) = \mathsf{EG}$ it follows that $(\prod_D \mathbb{G}_n)^+$ does not belong to EG. ❑

Nevertheless, the converse of Fine's theorem may fail be true in general, in many interesting cases it does hold — we refer to Goldblatt, Hodkinson & Venema [46] for a state of the art survey. Note that it is still an open problem whether every *finitely axiomatizable* canonical variety is elementarily generated.

Finally, recent work has put Fine's result in a wider algebraic context. We formulate the following theorem for Boolean algebras with operators, but in fact, it holds in a much wider setting, see for instance Gehrke & Harding [28].

THEOREM 98. *Let K be a class of Boolean algebras with τ-operators which is closed under taking ultraproducts and canonical extensions. Then the variety generated by K is canonical.*

Proof. Let $\mathbb{A}$ be in the variety generated by K; we will show $\mathbb{A}^\sigma$ to belong to $\mathsf{Var}(\mathsf{K})$ as well. By Tarski's 'HSP'-theorem, there is a family $\{\mathbb{B}_i \mid i \in I\} \subseteq \mathsf{K}$, and an algebra $\mathbb{B}$ such that $\mathbb{A} \twoheadleftarrow \mathbb{B} \rightarrowtail \prod_I \mathbb{B}_i$. Then it follows from two times two applications of Theorem 76 that $\mathbb{A}^\sigma \twoheadleftarrow \mathbb{B}^\sigma \rightarrowtail (\prod_I \mathbb{B}_i)^\sigma$, so it suffices to show that $(\prod_I \mathbb{B}_i)^\sigma$ belongs to $\mathsf{Var}(\mathsf{K})$. However, we may infer from Theorem 77 and Theorem 76(5) that

$$\left(\prod\nolimits_I \mathbb{B}_i\right)^\sigma \cong \prod_{D \in Uf(I)} \left(\prod\nolimits_D \mathbb{A}_i\right)^\sigma . \tag{24}$$

But by the assumptions on K, each algebra $(\prod_D \mathbb{A}_i)^\sigma$ belongs to K, and so the product (24) is in $\mathsf{P}(\mathsf{K}) \subseteq \mathsf{Var}(\mathsf{K})$, as required. ❑

From the above result we can derive Fine's Theorem as follows. Suppose that C is a frame class, closed under taking ultraproducts; for instance, let C be elementary. Then consider the class $\mathsf{SCm}(\mathsf{C})$ of sub-complex algebras over C. This class can be shown to be closed under taking ultraproducts as a corollary of Proposition 78, and closed under taking canonical extensions as a corollary of Theorem 76 and Theorem 90 in Chapter 5 of this volume. Application of Theorem 98 then yields the desired result.

6.3 Interpolation

In the last part of this section we discuss another fundamental property of logics: interpolation. Interpolation is important for applications because it allows reasoning systems to be set up in a modular way. Since we have confined our attention to logics in the form of sets of theorems, the version of interpolation that we will consider here is the following.

DEFINITION 99. A modal logic L has the *local* or *Craig interpolation property* if for every two formulas φ and ψ such that $\vdash_L \varphi \to \psi$ there is an *interpolant*, that is, a formula χ with $\vdash_L \varphi \to \chi$ and $\vdash_L \chi \to \psi$ and such that each variable of χ occurs both in φ and in ψ.

The algebraic counterpart of interpolation involves the notion of *amalgamation.*

DEFINITION 100. Let K be a class of algebras.

A *V-formation* in K is a quintuple, presented as $\mathbb{B}_1 \overset{e_1}{\leftarrowtail} \mathbb{B}_0 \overset{e_1}{\rightarrowtail} \mathbb{B}_2$, and consisting of three algebras $\mathbb{B}_0$, $\mathbb{B}_1$ and $\mathbb{B}_2$ in K, linked by two embeddings e_0 and e_1. An *amalgam* of this V-formation is a formation $\mathbb{B}_1 \overset{f_1}{\rightarrowtail} \mathbb{B}_{12} \overset{f_1}{\leftarrowtail} \mathbb{B}_2$ such that $f_1 \circ e_1 = f_2 \circ e_2$. Such a amalgam is a *superamalgam* if for all distinct i and j, and all $b_i \in B_i$ and $b_j \in B_j$: $f_i(b_i) \leq_{12} f_j(b_j)$ only if there is some $b_0 \in B_0$ with $b_i \leq_i e_i(b_0)$ and $e_j(b_0) \leq_j b_j$.

$$\begin{tikzcd}
& \mathbb{B}_{12} & \\
\mathbb{B}_1 \arrow[ur, dotted, "f_1"] & & \mathbb{B}_2 \arrow[ul, dotted, "f_2"'] \\
& \mathbb{B}_0 \arrow[ul, "e_1"] \arrow[ur, "e_2"'] &
\end{tikzcd}$$

K is said to have the *(super)amalgamation property* if every V-formation in K has a (super)amalgam in K.

In words, an amalgam is a superamalgam if whenever a $\mathbb{B}_i$-element is smaller (in $\mathbb{B}_{12}$) than a $\mathbb{B}_j$-element, then this is *witnessed* by a $\mathbb{B}_0$-element. The basic result connecting interpolation and amalgamation is from Maksimova [83].

THEOREM 101. *Let L be a normal modal τ-logic. Then L has the local interpolation property if and only if* $\mathsf{BAO}_\tau(L)$ *has superamalgamation.*

Proof. Fix L. In the proof of this theorem we will frequently consider Lindenbaum-Tarski algebras for L over various distinct sets of variables. Our notational convention will be that these sets of variables will always be called X_0, X_1, X_2 and X_{12}, with $X_0 = X_1 \cap X_2$ and $X_{12} = X_1 \cup X_2$; that $\mathbb{F}_i$ denotes the Lindenbaum-Tarski algebra over X_i; that $[\varphi]_i$ denotes the equivalence class of the formula φ under the L-equivalence relation $\equiv_L$ *within* the set $Fma(X_i)$; and, finally, if $X_i \subseteq X_j$, that $\iota_{i,j}$ denotes the map given by $[\varphi]_i \mapsto [\varphi]_j$. We leave it for the reader to verify that $\iota_{i,j}$ is an *embedding* of $\mathbb{F}_i$ into $\mathbb{F}_j$.

It is not hard to prove that L has local interpolation iff for all sets X_1 and X_2 of variables, the formation $\mathbb{F}_1 \overset{\iota_{1,12}}{\rightarrowtail} \mathbb{F}_{12} \overset{\iota_{2,12}}{\leftarrowtail} \mathbb{F}_2$ is a superamalgam of the V-formation $\mathbb{F}_1 \overset{\iota_{0,1}}{\leftarrowtail} \mathbb{F}_0 \overset{\iota_{0,2}}{\rightarrowtail} \mathbb{F}_2$. This observation already takes care of the direction from right to left of the theorem.

For the other direction we have to work harder. Consider a V-formation $\mathbb{B}_1 \overset{e_1}{\leftarrowtail} \mathbb{B}_0 \overset{e_1}{\rightarrowtail} \mathbb{B}_2$ in $\mathsf{BAO}_\tau(L)$. Without loss of generality we may assume that $B_0 = B_1 \cap B_2$. Wanting to use local interpolation of L to find a superamalgam of this V-formation, we translate the V-formation into syntax.

With $X_i := \{x_b \mid b \in B_i\}$ for each $i \in \{0, 1, 2\}$, let $\beta_i : \mathbb{F}_i \to \mathbb{B}_i$ be the unique homomorphism determined by the map $[x_b] \mapsto b$, cf. the picture. Clearly each β_i is surjective, whence by universal algebra, each $\mathbb{B}_i$ is isomorphic to the algebra $\mathbb{F}_i / \ker(\beta_i)$. Let M_i be the modal filter of $\mathbb{F}_i$ associated with the congruence $\ker(\beta_i)$ (as in Theorem 29), and let M be the modal filter of $\mathbb{F}_{12}$ generated by the union of M_1 and M_2, or, to be more precise, by the set $\iota_{1,12}[M_1] \cup \iota_{1,12}[M_1]$. We claim that the algebra $\mathbb{F}_{12}/\Pi_M$ is the required superamalgam, with Π_M the congruence associated with M, again, as in Theorem 29.

$$\begin{tikzcd}
 & \mathbb{B}_1 \arrow[rr, dotted] & & \mathbb{F}_{12}/\Pi_M \\
\mathbb{F}_1 \arrow[ur, "\beta_1"] \arrow[rr, "\iota_{1,12}"] & & \mathbb{F}_{12} \arrow[ur] & \\
 & \mathbb{B}_0 \arrow[uu, "e_1"] \arrow[rr, "e_2"] & & \mathbb{B}_2 \arrow[uu, dotted] \\
\mathbb{F}_0 \arrow[uu, "\iota_{0,1}"] \arrow[ur, "\beta_0"] \arrow[rr, "\iota_{0,2}"] & & \mathbb{F}_2 \arrow[uu, "\iota_{2,12}"] \arrow[ur, "\beta_2"] &
\end{tikzcd}$$

Proving this, the crucial observation is that $[\varphi]_{12}$ belongs to M iff there are formulas $\varphi_1 \in Fma(X_1)$ and $\varphi_2 \in Fma(X_2)$ such that $\vdash_L (\varphi_1 \wedge \varphi_2) \to \varphi$, and $[\varphi_i]_i \in M_i$ for $i = 1, 2$. From this, using local interpolation, it may be derived that for formulas $\psi_1 \in Fma(X_1)$ and $\psi_2 \in Fma(X_2)$, we have $[\psi_1 \to \psi_2]_{12} \in M$ iff there is a $\chi \in Fma(X_0)$ such that $[\psi_1 \to \chi]_1 \in M_1$ and $[\chi \to \psi_2]_2 \in M_2$. And from this the desired properties of $\mathbb{F}_{12}/\Pi_M$ follow almost immediately. ❑

This theorem can be applied to obtain a fairly general interpolation result for *canonical* modal logics that define nice frame classes. We need the following definition.

DEFINITION 102. Let $\mathbb{S}_1$ and $\mathbb{S}_2$ be two τ-frames. The direct product $\mathbb{S}_1 \times \mathbb{S}_2$ of these frames is the frame based on the Cartesian product $S_1 \times S_2$, with the relations defined coordinate-wise (for instance, in the case of a binary relation R, we put $R(s_1, s_2)(t_1, t_2)$ if $R_1 s_1 t_1$ and $R_2 s_2 t_2$). A subframe $\mathbb{Z}$ of $\mathbb{S}_1 \times \mathbb{S}_2$ is called a *zigzag product* of $\mathbb{S}_1$ and $\mathbb{S}_2$ if Z is a hereditary subset of the product frame on which the projection maps are surjective.

Clearly then zigzag products are substructures of direct products. A different perspective is that zigzag products of $\mathbb{S}_1$ and $\mathbb{S}_2$ are given by those bisimulations Z between $\mathbb{S}_1$ and $\mathbb{S}_2$ that are *full*, i.e., have domain S_1 and range S_2.

As an example of a zigzag product, consider two surjective bounded morphisms θ_1, θ_2 with $\theta_i : \mathbb{S}_i \twoheadrightarrow \mathbb{S}_0$. Then the frame $\mathbb{E}(\theta_1, \theta_2)$ based on the set $\{(s_1, s_2) \in S_1 \times S_2 \mid \theta_1(s_1) = \theta_2(s_2)\}$ is a zigzag product of $\mathbb{S}_1$ and $\mathbb{S}_2$. We call this the zigzag product *induced* by θ_1 and θ_2.

The following theorem, which is a generalization from Marx [84] of a result by Németi [87], is useful for proving that a canonical logic has interpolation.

THEOREM 103. *Let* K *be a class of Boolean algebras with* τ*-frames, and* C *a class of* τ*-frames such that* $\mathsf{Cst}(\mathsf{K}) \subseteq \mathsf{C}$*,* $\mathsf{Cm}(\mathsf{C}) \subseteq \mathsf{K}$*, and* C *is closed under taking zigzag products. Then* K *has the superamalgamation property.*

Proof. Suppose that K and C have the listed properties, and consider a V-formation

$$\mathbb{B} \stackrel{\alpha}{\leftarrowtail} \mathbb{A} \stackrel{\alpha'}{\rightarrowtail} \mathbb{B}'. \tag{25}$$

It follows from Theorem 76(3) that $\mathbb{B}_\bullet \stackrel{\alpha_\bullet}{\twoheadrightarrow} \mathbb{A}_\bullet \stackrel{\alpha'_\bullet}{\twoheadleftarrow} \mathbb{B}'_\bullet$. Now let $\mathbb{E}$ be the zigzag product of $\mathbb{B}_\bullet$ and $\mathbb{B}'_\bullet$ induced by the bounded morphisms $\alpha_\bullet$ and $\alpha'_\bullet$. Note that $\mathbb{E}$ belongs to $\mathbb{C}$ by the listed closure properties. Letting π and π' be the (surjective!) bounded morphisms from $\mathbb{E}$ onto $\mathbb{B}_\bullet$ and $\mathbb{B}'_\bullet$, respectively, we see that $\mathbb{B}_\bullet \stackrel{\pi}{\twoheadleftarrow} \mathbb{E} \stackrel{\pi'}{\twoheadrightarrow} \mathbb{B}'_\bullet$. It then follows from Theorem 76(2) and Theorem 54 that

$$\mathbb{B} \stackrel{\widehat{\cdot}}{\rightarrowtail} \mathbb{B}^\sigma \stackrel{\pi^+}{\rightarrowtail} \mathbb{E}^+ \stackrel{\pi'^+}{\leftarrowtail} \mathbb{B}'^\sigma \stackrel{\widehat{\cdot}}{\leftarrowtail} \mathbb{B}' \tag{26}$$

We claim that in fact, (26) is a superamalgam of (25), but leave further proof details for the reader. ❑

As a corollary of this theorem, suppose that Γ is a set of canonical formulas defining an elementary frame class that is closed under taking direct products and substructures — for instance, Γ corresponds to a set of universal Horn sentences. Then $\mathbf{K}_\tau.\Gamma$ has Craig interpolation.

Chapter 8 of this volume contains more information on interpolation. Related properties, such as Beth definability, also have algebraic characterizations; for details we refer to HOOGLAND [59].

7 CASE STUDY: CANONICAL EQUATIONS

7.1 Introduction

In this section we address the question, which equations are *canonical*, that is, remain valid when we move from a BAO $\mathbb{A}$ to its canonical embedding algebra $\mathbb{A}^\sigma$. In other words, we are interested in properties that move to certain *superalgebras*.

Earlier on we defined $\mathbb{A}^\sigma$ via a concrete construction, namely, as the 'double dual' $(\mathbb{A}_\bullet)^+$: the complex algebra of the ultrafilter frame of $\mathbb{A}$. In this section we will take a rather more abstract approach in which we first consider the canonical extension $\mathbb{B}^\sigma$ of the Boolean reduct $\mathbb{B}$ of $\mathbb{A}$; this $\mathbb{B}^\sigma$ is not constructed but axiomatically characterized as the (modulo isomorphism) unique completion of $\mathbb{B}$ in which $\mathbb{B}$ is *dense* and *compact*. Then the property of density suggests a canonical way to extend the interpretation of the operators on $\mathbb{B}$ to operations on $\mathbb{B}^\sigma$, thus providing the canonical extension $\mathbb{A}^\sigma$ of $\mathbb{A}$.

This algebraic method originates with the original BAO paper Jónsson and Tarski [70], but it differs from the duality-based approach of for instance Sambin & Vaccaro [100] that modal logicians usually take. In order to compare the two approaches, consider the following picture, introducing the four main characters of this story:

$$\begin{array}{|c|c|}\hline \mathbb{A} & \mathbb{A}_* \\ \hline \mathbb{A}^\sigma & \mathbb{A}_\bullet \\ \hline \end{array} \tag{27}$$

In the duality-based approach, one compares the frame (frame-based) structures on the right hand side of the picture, cf. the discussion on the notion of *persistence* in Chapter 5 of this volume, while the algebraic method stays purely on the left hand side, basically by encoding the relevant topological concepts into the algebraic framework. An advantage of the duality-based method is that it allows a treatment of canonicity in tandem with correspondence; on the other hand, the more abstract and 'duality-free' nature of the other approach enables its transportation to a much wider setting than that of canonical extensions of Boolean algebras with operators. In recent years, the algebraic approach has proven its use for *lattices* expanded with *arbitrary* operations, and has been applied to other kinds of completions than the perfect extension of Jónsson and Tarski.

Our exposition of this algebraic approach in the sections 7.2 to 7.5 is based on work by Jónsson [69], Gehrke & Jónsson [30, 31, 32] and Gehrke & Harding [28], while the very similar approach by Ghilardi & Meloni [34] should also be mentioned here. In our presentation we try to be as general as possible while keeping the section self-contained, and staying within the framework of Boolean algebras. Almost all our formulations apply to lattice-ordered algebras as well, however; we will come back to this issue towards the end of the section when we discuss further generalizations of the theory presented here.

For an outline, recall that the validity of equations can be formulated using term functions:

$$\mathbb{A} \models s \approx t \text{ iff } s^{\mathbb{A}} = t^{\mathbb{A}}. \tag{28}$$

Hence, for the canonical extension of $\mathbb{A}$, we find that

$$\mathbb{A}^\sigma \models s \approx t \text{ iff } s^{\mathbb{A}^\sigma} = t^{\mathbb{A}^\sigma}. \tag{29}$$

Now suppose that we have developed a canonical way to extend an n-ary map $f : A^n \to A$ to an n-ary map $f^\sigma : (A^\sigma)^n \to A^\sigma$; it then immediately follows from (28) that

$$\mathbb{A} \models s \approx t \text{ only if } (s^{\mathbb{A}})^\sigma = (t^{\mathbb{A}})^\sigma. \tag{30}$$

Hence, in case s and t are *stable* on $\mathbb{A}$, that is, if $(s^{\mathbb{A}})^\sigma = s^{\mathbb{A}^\sigma}$ and $(t^{\mathbb{A}})^\sigma = t^{\mathbb{A}^\sigma}$, then we may infer from $\mathbb{A} \models s \approx t$ that $\mathbb{A}^\sigma \models s \approx t$. This motivates a careful analysis of the relation between the functions $s^{\mathbb{A}^\sigma}$ (the term function of s in $\mathbb{A}^\sigma$) and $(s^{\mathbb{A}})^\sigma$ (the extension to $\mathbb{A}^\sigma$ of the term function $s^{\mathbb{A}}$). This analysis crucially involves the question, which f

and g satisfy $(f \circ g)^\sigma = f^\sigma \circ g^\sigma$. We will see that such cases of $(\cdot)^\sigma$ distributing over function composition admit a satisfactory explanation in terms of 'matching continuity properties' of the maps f^σ and g^σ. For this purpose we will endow canonical extensions of Boolean algebras with topological structure.

7.2 Canonical extensions of Boolean algebras

In this section we define the canonical extension of a Boolean algebra $\mathbb{B}$ as the unique *completion* of $\mathbb{B}$ in which $\mathbb{B}$ is *dense* and *compact*. We introduce these notions one by one.

A Boolean algebra $\mathbb{C}$ is a *completion* of a Boolean algebra $\mathbb{B}$ if $\mathbb{C}$ is complete and $\mathbb{B}$ is a subalgebra of $\mathbb{C}$. If $\mathbb{C}$ agrees with $\mathbb{B}$ on *all* meets and joins, then we call $\mathbb{C}$ a *regular* completion of $\mathbb{B}$, but in general we do not require completions to be regular. Thus the notation $\vee$ for finite joins is unambiguous, but not so for infinite joins. Our convention will be that $\bigvee X$ always denotes $\bigvee^{\mathbb{C}} X$, that is, the join taken in the completion.

For an example of a completion, consider a field of sets $\mathbb{S} = \langle S, A \rangle$ and note that the power set algebra $\mathbb{P}S$ is a completion of $\mathbb{S}^*$.

Before we define the concept of density, we introduce some preliminary notions. Given a completion $\mathbb{C}$ of the Boolean algebra $\mathbb{B}$, we call an element $c \in C$ *closed* (*open*) if c is the meet (join, respectively) in $\mathbb{C}$ of elements in B. We let $K_{\mathbb{C}}(B)$ and $O_{\mathbb{C}}(B)$ denote the collections of closed and open elements, respectively. Objects (such as the elements of B) that are both closed and open are called *clopen*. This terminology is in accordance with the topological perspective on fields of sets as in Remark 60. In the sequel, we may write $K_{\mathbb{C}}$, $K(B)$, or even K, instead of $K_{\mathbb{C}}(B)$, if the suppressed details are clear from context; and similarly for the set $O_{\mathbb{C}}(B)$.

We say that $\mathbb{B}$ is *meet-dense* in $\mathbb{C}$ if $K_{\mathbb{C}}(B) = C$, *join-dense* if $O_{\mathbb{C}}(B) = C$, and *dense* if $K_{\mathbb{C}}(O_{\mathbb{C}}(B)) = O_{\mathbb{C}}(K_{\mathbb{C}}(B)) = C$. In words, A is dense in $\mathbb{C}$ if every element of C is both a meet of open elements, and a join of closed elements. As a simple example of join-density, note that a Boolean algebra is atomic iff the collection of atoms forms a join-dense set. Building on this, we leave it as an exercise for the reader to verify that a field of sets $\mathbb{S} = \langle S, A \rangle$ is differentiated iff $\mathbb{S}^*$ is dense in $\mathbb{P}S$.

Now we turn to the notion of compactness. Given a completion $\mathbb{C}$ of the Boolean algebra $\mathbb{B}$, we say that $\mathbb{B}$ is *compact in* $\mathbb{C}$ if for all sets X and Y of closed and open elements, respectively, $\bigwedge X \leq \bigvee Y$ implies the existence of finite subsets $X_0 \subseteq X$, $Y_0 \subseteq Y$ such that $\bigwedge X_0 \leq \bigvee Y_0$. An alternative (but equivalent) characterization of compactness is that, for any closed p and open u,

$$p \leq u \text{ only if } p \leq b \leq u \text{ for some } b \in B,$$

as can easily be verified. Also note that, again, our definition of compactness coincides with standard topological terminology; this easily follows from the observation that for any pair C, U of collections of subsets of a set S, we have $\bigcap C \subseteq \bigcup U$ iff $S \subseteq \bigcup U \cup \bigcup \{\sim_S c \mid c \in C\}$.

We are now ready to define canonical extensions.

DEFINITION 104. A completion $\mathbb{C}$ of the Boolean algebra $\mathbb{B}$ is called a *canonical extension of* $\mathbb{B}$ if $\mathbb{B}$ is both compact and dense in $\mathbb{C}$.

It is in fact a rather strong property for one Boolean algebra to be the canonical extension of another. To start with, every Boolean algebra has a *unique* canonical extension.

THEOREM 105. *Let $\mathbb{B}$ be some Boolean algebra. Then*

1. *(existence) $\mathbb{B}$ has a canonical extension;*
2. *(unicity) Any two canonical extensions of $\mathbb{B}$ are isomorphic via a unique isomorphism that restricts to the identity on B.*

Proof. Recall from the topological duality that $\mathbb{B}_* = \langle Uf\mathbb{B}, \widehat{B}\rangle$ is a differentiated and compact field of sets. By the comments made above it should be clear that $\mathbb{P}(Uf\mathbb{B})$ is a canonical extension of $\mathbb{B}$.

For unicity, suppose that $\mathbb{C}$ is a canonical extension of $\mathbb{B}$. We leave it as an exercise for the reader to verify that, by compactness, the map $F \mapsto \bigwedge F$ forms a dual (that is, order-reversing) isomorphism between the lattice $\langle Fi(\mathbb{B}), \subseteq\rangle$ and the induced ordering on the set $K(B)$ of closed elements. Its inverse is given by the map $p \mapsto \{a \in B \mid a \geq p\}$. Similarly, there is a dual isomorphism between the lattice of ideals of $\mathbb{B}$, and the induced ordering of the open elements. Also, we have for p closed and u open, that $p \leq u$ iff there is an $a \in B$ with $p \leq a \leq u$, and that $u \leq p$ iff $a \leq b$ for all a and b in A with $a \leq u$ and $p \leq b$. In other words, by *compactness* the induced poset on the set $K \cup O$ of closed or open elements is completely determined by the ordering of $\mathbb{B}$. This suffices to prove the theorem, since by *density*, the elements of $\mathbb{C}$ can be identified with the pairs (L, U) of subsets of C such that L is the collection of closed lower bounds of U, and U is the collection of open upper bounds of L. Summarizing, we see that together, compactness and density completely fix the order relation of the canonical extension. ❑

The above theorem justifies our speaking of 'the' canonical extension of a Boolean algebra $\mathbb{B}$; this algebra will be denoted as $\mathbb{B}^\sigma$. Furthermore, we need the following facts.

PROPOSITION 106. *Let $\mathbb{C}$ be a canonical extension of the Boolean algebra $\mathbb{B}$. Then*

1. *$B = K(B) \cap O(B)$; that is, B coincides with the set of clopen elements of $\mathbb{C}$;*
2. *the set $K(B)$ forms a sublattice of $\mathbb{C}$ which is closed under taking infinitary meets;*
3. *$\mathbb{C}$ is atomic and $At\mathbb{C} \subseteq K(B)$; that is, all atoms are closed.*

We leave the proof of this proposition to the reader; note that by Theorem 105, it suffices to restrict attention to the double dual $\mathbb{P}(Uf\mathbb{B})$ of $\mathbb{B}$. For instance, part (3) follows almost immediately from the identification of atoms of $\mathbb{P}(Uf\mathbb{B})$ with ultrafilters of $\mathbb{B}$.

As a last introductory remark, we note that canonical extensions interact well with finite products and order duals. Concerning the latter notion, recall that the *order dual* of a Boolean algebra $\mathbb{B} = \langle B, \top, \bot, -, \wedge, \vee\rangle$ is the structure $\mathbb{B}^\partial = \langle B, \bot, \top, -, \vee, \wedge\rangle$. The fact, that $\mathbb{B}^\partial$ is a Boolean algebra as well, enables us to shorten quite a lot of definitions and proofs by referring to the *principle of order duality*: Every fact concerning Boolean algebras remains valid after swapping $\top$ with $\bot$, $\wedge$ with $\vee$, etc.

PROPOSITION 107. *Let $\mathbb{B}_1, \ldots, \mathbb{B}_n$ be Boolean algebras. Then*

1. *$(\mathbb{B}_1 \times \cdots \times \mathbb{B}_n)^\sigma \cong \mathbb{B}_1^\sigma \times \cdots \times \mathbb{B}_n^\sigma$;*
2. *$(\mathbb{B}^\partial)^\sigma \cong (\mathbb{B}^\sigma)^\partial$;*

Proof. Both statements can be proved on the basis of Theorem 76. As intermediate steps, one can prove facts like $K(\mathbb{B}_1) \times \cdots \times \mathbb{B}_n) = K(\mathbb{B}_1) \times \cdots \times K(\mathbb{B}_n)$ and $K(\mathbb{B}^\partial) = O(\mathbb{B})$. ❑

7.3 *Extending maps to the canonical extension*

In the introduction to this section we saw that in order to investigate the canonicity of an equation $s \approx t$, it is useful to define extensions of the term functions on a BAO to maps on the canonical extension of the BAO. But in fact, there are canonical ways to extend an *arbitrary* map between two Boolean algebras $\mathbb{A}$ and $\mathbb{B}$, to a map between $\mathbb{A}^\sigma$ and $\mathbb{B}^\sigma$. This general definition will be discussed at the end of this section — for the time being we will confine ourselves to extensions of *monotone* maps.

The easiest way to understand these definitions is to break them down in two steps. For a start, the definition of closed and open elements suggests the following extension of $f : \mathbb{A} \to \mathbb{B}$ to a map $\overline{f}$ defined on $K(A) \cup O(A)$:

$$\begin{array}{rcll} \overline{f}(p) & := & \bigwedge\{f(a) \mid p \leq a \in A\} & \text{for } p \in K(A), \\ \overline{f}(u) & := & \bigvee\{f(a) \mid u \geq a \in A\} & \text{for } u \in O(A). \end{array} \tag{31}$$

Note that this is a correct definition because $K \cap O = A$ by Proposition 106(1), that $\overline{f}(a) = f(a)$ for $a \in A$ by monotonicity of f, and that $\overline{f}$ itself is also order preserving.

Now for the second step of the construction. The fact that every element is both the join of the closed elements below it, and the meet of the opens above it, suggests *two* ways to proceed:

$$\begin{array}{rcl} f^\sigma(x) & := & \bigvee\{\overline{f}(p) \mid x \geq p \in K(A)\}, \\ f^\pi(x) & := & \bigwedge\{\overline{f}(u) \mid x \leq u \in O(A)\}. \end{array} \tag{32}$$

The maps f^σ and f^π are called the *lower* and *upper extension* of f, respectively.

Let us first gather some basic facts concerning these definitions. The following proposition says that the names 'lower', 'upper', and 'extension' are well chosen.

PROPOSITION 108. *Let $f : \mathbb{A} \to \mathbb{B}$ be a monotone map between Boolean algebras. Then*

1. *both f^σ and f^π extend f;*

2. *$f^\sigma \leq f^\pi$, with equality holding on the closed and on the open elements.*

Proof. The first statement is immediate by the definitions and the monotonicity of $\overline{f}$. For the second statement, take, for $x \in A^\sigma$, a closed $p \leq x$ and an open $u \geq x$. By compactness there is an $a \in [p, u] \cap A$. This element satisfies $\overline{f}(p) \leq f(a) \leq \overline{f}(u)$ by definition of $\overline{f}$; hence $f^\sigma(x) \leq f^\pi(x)$ by definition of f^σ and f^π. Finally, for closed p we may derive from the first part of the proposition that $f^\pi(p) \leq \overline{f}(p)$, and from the monotonicity of $\overline{f}$ that $\overline{f}(p) = f^\sigma(p)$. Thus we obtain the desired equality $f^\sigma = f^\pi$ on K. The result for opens follows by order duality. ❑

Maps for which the lower and upper extension coincide are obviously of interest.

DEFINITION 109. A monotone map f between Boolean algebras is called *smooth* if $f^\sigma = f^\pi$.

EXAMPLE 110. As a first example of a smooth operation, consider the *global modality* g on a Boolean algebra $\mathbb{B}$, given by $g(\bot) = \bot$ while $g(b) = \top$ for $b > \bot$, see Definition 131. It is easy to see that $\overline{g}$ satisfies these conditions as well, whence it is equally easy to infer that both g^σ and g^π coincide with the global modality of $\mathbb{B}^\sigma$; smoothness is then immediate. Similarly, one can prove that the *meet* and *join* operations of $\mathbb{B}$ are smooth, and that their extensions coincide with the meet and the join of $\mathbb{B}^\sigma$, respectively.

For an operation that is not smooth, consider the composition of the global modality with the meet operation, i.e., the map $f : B^2 \to B$ given by $f(a, b) = \bot$ if $a \wedge b = \bot$, while $f(a, b) = \top$ otherwise. Now if $\mathbb{B}$ is infinite, then $\mathbb{B}^\sigma$ must contain some element c which is closed but not open; a straightforward verification shows that for such a c, we have that $f^\sigma(c, -c) = \bot$, while $f^\pi(c, -c) = \top$. This shows that not even operators are smooth.

While it may not be the case that the lower and the upper extension agree in all cases, both kinds of extensions generally display good behavior; often they even improve on the original map. For the definitions of the notions mentioned in the theorem below, see Definition 15 and 40.

PROPOSITION 111. *Let $f : \mathbb{A} \to \mathbb{B}$ be a map between Boolean algebras. Then*

1. *if f is monotone then so is f^σ;*
2. *if f is an operator then f^σ is a complete operator;*
3. *if f is additive or multiplicative then f is smooth.*

Proof. The proof of the first statement is easy and hence omitted, while we postpone the proof of the last statement (it is in fact a rather straightforward consequence of the Propositions 116 and 117). For the remaining part, we need to show that if f is normal and additive in each coordinate, then f^σ is normal and completely additive in each coordinate. Leaving the easy proof for normality as an exercise for the reader, concerning additivity, we will prove that if $f : \mathbb{A}_0 \times \mathbb{A}_1 \to \mathbb{B}$ is additive in its first coordinate and monotone in its second, then f^σ preserves all non-empty joins in its first coordinate.

Fix elements $x_0 \in A_0^\sigma$ and $x_1 \in A_1^\sigma$. By atomicity of $\mathbb{B}^\sigma$, and monotonicity of f^σ, it suffices to prove, for an arbitrary atom p of $\mathbb{B}^\sigma$:

$$p \leq f^\sigma(x_0, x_1) \text{ only if there is a } q \in At_0 \text{ with } p \leq f^\sigma(q, x_1), \tag{33}$$

where At_0 denotes the set of atoms in $\mathbb{A}_0^\sigma$ below x_0. Note that since $f^\sigma(x_0, x_1) = \bigvee\{f^\sigma(c_0, c_1) \mid x_i \geq c_i \in K(A_i)\}$ we may safely assume that both x_0 and x_1 are closed.

Now suppose for contradiction that (33) fails. Then for some atom p of $\mathbb{B}^\sigma$ we have $p \leq f^\sigma(x_0, x_1)$ while for each $q \in At_0$ there are, by definition of f^σ, elements $a_{q,0} \in A_0$ above q and $a_{q,1} \in A_1$ above x_1, such that $p \not\leq f^\sigma(a_{q,0}, a_{q,1})$. It follows that $x_0 = \bigvee At_0 \leq \bigvee\{a_{q,0} \mid q \in At_0\}$, whence by compactness $x_0 \leq \bigvee\{a_{q,0} \mid q \in F\}$ for some *finite* set $F \subseteq At_0$.

Now observe that the join $a_0 = \bigvee\{a_{q,0} \mid q \in F\}$ is in A_0, and the meet $a_1 = \bigwedge\{a_{q,1} \mid q \in F\}$ is in A_1. Clearly $p \not\leq f^\sigma(a_{q,0}, a_1)$ for each $q \in F$; since p is an atom this means $p \not\leq \bigvee\{f^\sigma(a_{q,0}, a_1) \mid q \in F\} = f(a_0, a_1)$, where in the last identity we use the additivity of f in its first coordinate.

On the other hand, from $x_0 \leq a_0$ and $x_1 \leq a_1$ it follows that $f^\sigma(x_0, x_1) \leq f(a_0, a_1)$ which gives the desired contradiction. ❑

In the proof above we already used the fact that complete additivity of f^σ means that it is completely determined by its values on the *atoms* of $\mathbb{B}^\sigma$. Now recall that (in the concrete representation of) $\mathbb{B}^\sigma$, the atoms are nothing but the *ultrafilters* of $\mathbb{B}$. From this the following proposition is immediate.

PROPOSITION 112. *Let* $\mathbb{A}$ *be some Boolean algebra with* τ*-operators with underlying Boolean algebra* $\mathbb{B}$*. Then* $\mathbb{A}^\sigma := (\mathbb{A}_\bullet)^+$ *is isomorphic to the algebra* $\mathbb{B}^\sigma$ *expanded with the family* $\{(\nabla^\mathbb{A})^\sigma \mid \nabla \in \tau\}$ *of complete operators.*

This proposition, which can be summarized as '$\nabla^{\mathbb{A}^\sigma} = (\nabla^\mathbb{A})^\sigma$', will be used throughout the sequel, but always implicitly.

7.4 Composite maps

We now investigate the interaction between composing maps between Boolean algebras and taking their canonical extensions. That is, we will take a look at the relation between the maps $(gf)^\sigma$ and $g^\sigma f^\sigma$ for maps $f : \mathbb{A} \to \mathbb{A}'$, and $g : \mathbb{A}' \to \mathbb{A}''$. We are obviously eager to find cases in which we have $(gf)^\sigma = g^\sigma f^\sigma$, but also conditions under which one of the inequalities ($\leq$ or $\geq$) apply will turn out to be of interest. As we will see shortly, many of these conditions can naturally be described in *topological* terms.

For this purpose, we will introduce no less than six topologies on each set A^σ. Fortunately, these topologies can be neatly organized in two families, each consisting of an upper, a lower and a join topology. As a terminological convention, let us call a map between the algebras $\mathbb{A}^\sigma$ and $\mathbb{B}^\sigma$ (ρ, ρ')*-continuous*, if it is a continuous function between the topological spaces $\langle A^\sigma, \rho\rangle$ and $\langle B^\sigma, \rho'\rangle$.

The first family is that of the Scott topologies. Although these can already be defined on arbitrary partial orders, here we will only consider topologies on canonical extensions of BAOs. Recall that a subset D of a partial order is called *up-directed*, if every pair of elements of D has an upper bound in D.

DEFINITION 113. Given a Boolean algebra $\mathbb{B}$, call a subset U of B^σ *Scott open* if U is an up-set such that $U \cap D \neq \varnothing$ for every up-directed set D with $\bigvee D \in U$. The *Scott topology* is defined as the collection $\gamma^\uparrow$ of Scott open sets; the topology $\gamma^\downarrow$ is given by the principle of order duality, and we define $\gamma := \{U \cap V \mid U \in \gamma^\uparrow, V \in \gamma^\downarrow\}$ as the join of $\gamma^\uparrow$ and $\gamma^\downarrow$ in the lattice of topologies over B.

In practice it is sometimes easier to work with the *closed* sets in the Scott topology; these are precisely the down-sets of $\mathbb{C}$ that are closed under taking up-directed unions. From this observation one easily derives the (well-known) fact that a map between partial orders is Scott continuous (that is, $(\gamma^\uparrow, \gamma^\uparrow)$-continuous) iff it preserves up-directed joins. But this implies that a map is completely additive iff it is both additive and Scott continuous, which may help to explain the relevance of the Scott topologies for our purposes.

We now turn to the second family of topologies. Recall from Example 27 that for an arbitrary element b of a Boolean algebra $\mathbb{B}$, the sets $b{\uparrow}$ and $b{\downarrow}$ are defined as $b{\uparrow} = \{a \in B \mid b \leq a\}$ and $b{\downarrow} = \{a \in B \mid a \leq b\}$.

PROPOSITION 114. *For any Boolean algebra* $\mathbb{B}$, *the sets* $\sigma^{\uparrow} := \{p{\uparrow} \mid p \in K\}$ *and* $\sigma^{\downarrow} := \{u{\downarrow} \mid u \in O\}$ *both form a topology on* A^{σ}*; and so does the set* $\sigma := \{p{\uparrow} \cap u{\downarrow} \mid K \ni p \leq u \in O\}$, *which is in fact identical to the join* $\sigma^{\uparrow} \vee \sigma^{\downarrow}$ *in the lattice of topologies on* A^{σ}.

In the sequel, we will write $[p, u]$ for the *interval* between p and u, that is, $[p, u] = p{\uparrow} \cap u{\downarrow}$.

Proof. The fact that $\sigma^{\uparrow}$ is a topology follows from the fact that the set $K(A)$ is closed under finitary joins and arbitrary meets of $\mathbb{A}^{\sigma}$, see Proposition 106(2). ❑

REMARK 115. As suggested by notation, the topology σ is closely connected to the kind of inclusion of $\mathbb{B}$ in $\mathbb{B}^{\sigma}$. Let us just mention a couple of salient facts here. First, it is easy to see that the set $\{[a, b] \mid a, b \in B\}$ is a *basis* for σ. This reveals that the set B is topologically *dense* in σ, in the sense that every σ-open set contains an element of B. But also, B constitutes the collection of *isolated points* of σ — recall that a point x is isolated in a topology if the singleton $\{x\}$ is open. It is the latter two properties that make it possible to extend arbitrary maps between Boolean algebras to their extensions; we will come back to this at the end of this section.

The following proposition, which links the two topological families, will be crucial when it comes to finding the 'matching continuities' mentioned in the introduction.

PROPOSITION 116. *Let* $\mathbb{A}$ *be a Boolean algebra. Then* $\gamma^{\uparrow} \subseteq \sigma^{\uparrow}$, $\gamma^{\downarrow} \subseteq \sigma^{\downarrow}$ *and* $\gamma \subseteq \sigma$.

Proof. Confining ourselves to the first claim, it suffices to prove that $U = \bigcup\{p{\uparrow} \mid p \in U \cap K\}$ for an arbitrary Scott open set $U \subseteq A^{\sigma}$. The crucial observation here is that every $u \in U$ is the *up-directed* join of the closed elements below it. Further proof details are left to the reader. ❑

The following proposition is a first sign that these topologies can be useful.

PROPOSITION 117. *Let* $f : \mathbb{A} \to \mathbb{B}$ *be a monotone map between the Boolean algebras* $\mathbb{A}$ *and* $\mathbb{B}$. *Then*

1. f^{σ} *is the largest monotone* $(\sigma, \gamma^{\uparrow})$*-continuous extension of* f*;*

2. f *is smooth iff* f^{σ} *is* (σ, γ)*-continuous;*

3. *if* f *is an operator then* f^{σ} *is* $(\gamma^{\uparrow}, \gamma^{\uparrow})$*-continuous;*

4. *if* f *is additive then* f^{σ} *is* $(\sigma^{\downarrow}, \sigma^{\downarrow})$*-continuous.*

5. *if* f *is multiplicative then* f^{σ} *is* $(\sigma^{\uparrow}, \sigma^{\uparrow})$*-continuous.*

Proof. Concerning the first part of the proposition, we already know from Proposition 108 that f^{σ} is an extension of f. Now for $x \in A^{\sigma}$ take an arbitrary Scott open set $V \subseteq B^{\sigma}$ with $f^{\sigma}(x) \in V$. That is, $\bigvee\{f^{\sigma}(p) \mid x \geq p \in K(A)\} \in V$. Now it is easy to see that the collection $Q := \{f^{\sigma}(p) \mid x \geq p \in K(A)\}$ is up-directed, so $Q \cap V \neq \varnothing$. In other words, there is a closed $p \leq x$ with $f^{\sigma}(p) \in V$. But then by monotonicity of f^{σ} we have that $f^{\sigma}[p{\uparrow}] \subseteq V$. Since $x \in p{\uparrow} \in \sigma$ this suffices to prove that f^{σ} is $(\sigma, \gamma^{\uparrow})$-continuous, while by Proposition 111 it is monotone.

In order to show that f is the largest such map, take a monotone $(\sigma, \gamma^{\uparrow})$-continuous extension $g : A^{\sigma} \to B^{\sigma}$ of f, and suppose for contradiction that $g(x) \not\leq f^{\sigma}(x)$ for some $x \in A^{\sigma}$. By atomicity of $\mathbb{B}^{\sigma}$ there must be an *atom* p of $\mathbb{B}^{\sigma}$ which lies below $g(x)$, but not below $f^{\sigma}(x)$. Because $g(x) \in p{\uparrow} \in \gamma^{\uparrow}$, the continuity of g provides us with a $c \in K$ such that $c \leq x$ and $g[c{\uparrow}] \subseteq p{\uparrow}$. In other words, we find that $p \leq g(c)$ whence by monotonicity it follows that $p \leq g(a)$ for all $a \in A$ above c. But then by the fact that g extends f, and the definition of f^{σ}, we may infer that $p \leq f^{\sigma}(c)$. From this we obtain, as the required contradiction, that $p \leq f^{\sigma}(x)$.

For part (2), it follows from part (1) by order duality that f^{π} is the smallest monotone $(\sigma, \gamma^{\downarrow})$-continuous extension of f. Hence if f is smooth, then $f^{\sigma} = f^{\pi}$ is both $(\sigma, \gamma^{\uparrow})$- and $(\sigma, \gamma^{\downarrow})$-continuous, and hence, (σ, γ)-continuous. Conversely, if f^{σ} is (σ, γ)-continuous, then it is, a fortiori, $(\sigma, \gamma^{\downarrow})$-continuous. This implies, again by the order dual of part (1), that $f^{\pi} \leq f^{\sigma}$; but then we have equality because of Proposition 108(2).

Concerning part (3), if $f : \mathbb{A}^n \to \mathbb{A}$ is an operator then by Proposition 111(2), $f^{\sigma} : (\mathbb{A}^{\sigma})^n \to \mathbb{A}^{\sigma}$ is additive in each coordinate. From this it is straightforward to derive that f^{σ} preserves up-directed joins.

For part (4), suppose that $f : \mathbb{A} \to \mathbb{B}$ is additive, and take an arbitrary $\sigma^{\downarrow}$-open subset $u{\downarrow}$ of B^{σ}, that is, $u \in O(B)$. It follows by Proposition 111(2) that f^{σ} preserves all non-empty joins. From this one may derive that the set $(f^{\sigma})^{-1}[u{\downarrow}]$ is either empty, in which case it certainly belongs to σ, or else it is of the form $v{\downarrow}$, where $v = \bigvee (f^{\sigma})^{-1}[u{\downarrow}]$ satisfies $f^{\sigma}(v) \leq u$. In order to show that $v{\downarrow}$ is open in σ, it suffices to prove that v is an open element of $\mathbb{A}^{\sigma}$.

Consider an arbitrary closed element $p \leq v$; then $\bigwedge f[p{\uparrow} \cap A] = f^{\sigma}(p) \leq f^{\sigma}(v) \leq u$. Hence by compactness there is a finite set $F \subseteq p{\uparrow} \cap A$ such that $\bigwedge f[F] \leq u$. Putting $a_p := \bigwedge F$ we find that $a_p \in A$, $p \leq a_p$ and $a_p \leq v$ since $f(a_p) \leq \bigwedge f[F]$. Clearly then $v = \bigvee \{p \mid v \geq p \in K\} \leq \bigvee \{a_p \mid v \geq p \in K\} \leq v$ which shows that v is *identical* to the second join, and hence, open.

Finally, part (5) follows from part (4) by order duality. ❑

As we announced already in the introduction to this section, the following properties will be crucial in proving canonicity results further on. The reason for this lies in the observation that for some terms t, we may apply Proposition 118(2) by the fact that the term function $t^{\mathbb{A}^{\sigma}}$ in the canonical extension $\mathbb{A}^{\sigma}$ can be decomposed as $t^{\mathbb{A}^{\sigma}} = g^{\sigma} \circ f^{\sigma}$ where g^{σ} is $(\tau, \gamma^{\uparrow})$-continuous and f^{σ} is (σ, τ)-continuous, for some 'intermediate' topology τ. This is the principle of *matching continuities* that we mentioned in the introduction.

PROPOSITION 118. *Let $f : \mathbb{A} \to \mathbb{B}$ and $g : \mathbb{B} \to \mathbb{C}$ be monotone maps between the Boolean algebras $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$. Then*

1. $(gf)^{\sigma} \leq g^{\sigma} f^{\sigma}$*;*
2. $(gf)^{\sigma} \geq g^{\sigma} f^{\sigma}$ *whenever* $g^{\sigma} f^{\sigma}$ *is* $(\sigma, \gamma^{\uparrow})$*-continuous.*

Proof. Part (2) of the proposition is an immediate consequence of Proposition 117(1) since $g^{\sigma} f^{\sigma}$ is an extension of gf (and gf is monotone). Concerning part (1), we first show that $(gf)^{\sigma}(p) \leq g^{\sigma} f^{\sigma}(p)$ for closed p. Note that

$$\begin{aligned} (gf)^{\sigma}(p) &= \bigwedge \{gf(a) \mid p \leq a \in A\}, \\ g^{\sigma} f^{\sigma}(p) &= \bigwedge \{g(b) \mid f^{\sigma}(p) \leq b \in B\}. \end{aligned}$$

where the latter identity holds because $f^\sigma(p)$ is closed in $\mathbb{A}^\sigma$. Now take a $b \in B$ with $f^\sigma(p) \leq b$. As $f^\sigma(p) = \bigwedge\{f(a) \mid p \leq a \in A\}$ is a down-directed meet, compactness provides some $a \in A$ with $p \leq a$ and $f(a) \leq b$. Then $(gf)^\sigma(p) \leq gf(a) \leq gb$; and hence, $(gf)^\sigma(p) \leq g^\sigma f^\sigma(p)$.

Now we turn to arbitrary $x \in A^\sigma$. Note that

$$\begin{aligned} (gf)^\sigma(x) &= \bigvee\{(gf)^\sigma(p) \mid x \geq p \in K(A)\}, \\ g^\sigma f^\sigma(x) &= \bigvee\{g^\sigma(q) \mid f^\sigma(x) \geq q \in K(B)\}. \end{aligned}$$

Take an arbitrary $p \in K(A)$ with $p \leq x$; then $(gf)^\sigma(p) \leq g^\sigma f^\sigma(p)$, as we just saw. Since $f^\sigma(x) \geq f^\sigma(p) \in K(B)$, this shows that every joinand $(gf)^\sigma(p)$ of $(gf)^\sigma(x)$ is below some joinand $g^\sigma(q)$ of $g^\sigma f^\sigma(x)$. This suffices to prove the desired inequality. ❑

7.5 Canonical equations

Time to harvest. The key idea for proving canonicity results for an equation $s \approx t$ will be to use properties of the *term functions* $s^\mathbb{A}$ and $t^\mathbb{A}$. Recall that for a term $t(x_1, \ldots, x_n)$, the term function $t^\mathbb{A} : A^n \to A$ is inductively defined as follows:

$$\begin{aligned} x_i^\mathbb{A} &:= \pi_i^n, \\ (\heartsuit(t_1, \ldots, t_n))^\mathbb{A} &:= \heartsuit^\mathbb{A} \circ \langle t_1^\mathbb{A}, \ldots, t_n^\mathbb{A}\rangle. \end{aligned}$$

where $\pi_i^n : (a_1, \ldots, a_n) \mapsto a_i$ is the i-th *projection* function, and, for maps $f_1, \ldots, f_n : X \to Y$, the map $\langle f_1, \ldots, f_n\rangle : X \to Y^n$ is given by $\langle f_1, \ldots, f_n\rangle(x) = (f_1(x), \ldots, f_n(x))$.

In the context of canonical extensions the following definitions are crucial.

DEFINITION 119. A term t is *expanding* on an expanded Boolean algebra $\mathbb{A}$ if $(t^\mathbb{A})^\sigma \leq t^{\mathbb{A}^\sigma}$, *contracting* if $(t^\mathbb{A})^\sigma \geq t^{\mathbb{A}^\sigma}$, and *stable* if $(t^\mathbb{A})^\sigma = t^{\mathbb{A}^\sigma}$. We let these properties apply to classes of algebras in case they apply to all members of the class.

PROPOSITION 120. *Let s and t be two τ-terms, and K a class of τ-expanded Boolean algebras. If s is contracting and t is expanding on K, then the inequality $s \preceq t$ is canonical on K.*

Proof. Consider an algebra $\mathbb{A}$ in K such that $\mathbb{A} \models s \preceq t$. In other words, we have $s^\mathbb{A} \leq t^\mathbb{A}$, so that $(s^\mathbb{A})^\sigma \leq (t^\mathbb{A})^\sigma$. But then by the assumptions on s and t it follows that $s^{\mathbb{A}^\sigma} \leq (s^\mathbb{A})^\sigma \leq (t^\mathbb{A})^\sigma \leq t^{\mathbb{A}^\sigma}$, which shows that $\mathbb{A}^\sigma \models s \preceq t$. ❑

So which terms are contracting, and which ones are expanding? Here the topologies prove their value. Before moving on to these results, we need to get one technicality out of the way. Basically, the following proposition states that the product map $\langle f_1, \ldots, f_n\rangle$ behaves as well as one could hope for.

PROPOSITION 121. *Let $f_1, \ldots, f_n$ be monotone maps between the Boolean algebras $\mathbb{A}$ and $\mathbb{B}$. Then*

$$\langle f_1, \ldots, f_n\rangle^\sigma = \langle f_1^\sigma, \ldots, f_n^\sigma\rangle,$$

and for all $\rho, \rho' \in \{\gamma^\downarrow, \gamma^\uparrow, \gamma, \sigma^\downarrow, \sigma^\uparrow, \sigma\}$ it holds that

$$\langle f_1, \ldots, f_n\rangle^\sigma \textit{ is } (\rho, \rho')\textit{-continuous iff each } f_i^\sigma \textit{ is } (\rho, \rho')\textit{-continuous.}$$

We leave the rather tedious but not very difficult proof of this proposition to the reader, and move on to more interesting facts. First we associate topological properties with term functions.

PROPOSITION 122. *Let $\mathbb{A}$ be a τ-expanded Boolean algebra, and t a τ-term. Then*

1. *If $\mathbb{A}$ interprets all connectives in t as operators, then $t^{\mathbb{A}^\sigma}$ is $(\gamma^\uparrow, \gamma^\uparrow)$-continuous.*
2. *If $\mathbb{A}$ interprets all connectives in t as additive maps, then $t^{\mathbb{A}^\sigma}$ is $(\sigma^\downarrow, \sigma^\downarrow)$-continuous.*
3. *If $\mathbb{A}$ interprets all connectives in t as multiplicative maps, then $t^{\mathbb{A}^\sigma}$ is $(\sigma^\uparrow, \sigma^\uparrow)$-continuous.*

Proof. All three statements can be proved by a straightforward term induction, using the Propositions 117 and 121 for the induction step. For the induction base, note that the projection maps are both join- and meet preserving, and hence, their canonical extensions have all the continuity properties mentioned in the statements of this proposition. ❑

Here we arrive at the core of the algebraic approach towards the canonicity of equations. On the basis of the syntactic shape of some terms we can see whether it is expanding or stable. In Theorem 123 we give some sample results; observe that the key idea in the proof of part (3) is the principle of 'matching continuities' as described before Proposition 118.

THEOREM 123. *Let $\mathbb{A}$ be a τ-expanded Boolean algebra, and t a τ-term. Then*

1. *If $\mathbb{A}$ interprets all connectives in t as monotone maps, then t is expanding.*
2. *If $\mathbb{A}$ interprets all connectives in t as operators or dual operators, then t is stable.*
3. *If t is of the form $s(u_1, \ldots, u_n)$ such that $\mathbb{A}$ interprets all connectives in s as operators, and all connectives in each of the u_i as meet-preserving operations, then t is stable.*

Proof. Part (1) is proved by term induction. The base case is immediate from the definitions. For the inductive step, suppose that $t \equiv \nabla(t_1, \ldots, t_n)$, then

$$
\begin{array}{rcl}
(t^{\mathbb{A}})^\sigma & = & (\nabla^{\mathbb{A}} \circ \langle t_1^{\mathbb{A}}, \ldots, t_n^{\mathbb{A}} \rangle)^\sigma \\
& \leq & (\nabla^{\mathbb{A}})^\sigma \circ \langle t_1^{\mathbb{A}}, \ldots, t_n^{\mathbb{A}} \rangle^\sigma \\
& = & \nabla^{\mathbb{A}^\sigma} \circ \langle (t_1^{\mathbb{A}})^\sigma, \ldots, (t_n^{\mathbb{A}})^\sigma \rangle \\
& \leq & \nabla^{\mathbb{A}^\sigma} \circ \langle t_1^{\mathbb{A}^\sigma}, \ldots, t_n^{\mathbb{A}^\sigma} \rangle \\
& = & t^{\mathbb{A}^\sigma}.
\end{array}
$$

Here the first and last step are by definition, the second step is by Proposition 118(1) and monotonicity, the third step is by definition of $\nabla^{\mathbb{A}^\sigma} = (\nabla^{\mathbb{A}})^\sigma$ and by Proposition 121, and the fourth step is by the inductive hypothesis and the monotonicity of $\nabla^{\mathbb{A}^\sigma}$.

For part (2) and (3) it suffices to prove that $t^{\mathbb{A}^\sigma} \leq (t^{\mathbb{A}})^\sigma$, since the opposite inequality holds by part (1). In the case of part (2) this follows from a straightforward induction, whereas for part (3) we need the principle of matching topologies.

Let t be as described in part (3), then

$$t^{\mathbb{A}^\sigma} = s^{\mathbb{A}^\sigma} \circ \langle u_1^{\mathbb{A}^\sigma}, \ldots, u_n^{\mathbb{A}^\sigma} \rangle = (s^{\mathbb{A}})^\sigma \circ \langle (u_1^{\mathbb{A}})^\sigma, \ldots, (u_n^{\mathbb{A}})^\sigma \rangle$$

with the second identity holding by part (2). Also, note that by Proposition 117, the term function $s^{\mathbb{A}^\sigma}$ is $(\gamma^\uparrow, \gamma^\uparrow)$-continuous, and each $u_i^{\mathbb{A}^\sigma}$ is $(\sigma^\uparrow, \sigma^\uparrow)$-continuous. From this we infer by Proposition 121 that the map $\langle (u_1^{\mathbb{A}})^\sigma, \ldots, (u_n^{\mathbb{A}})^\sigma \rangle = \langle u_1^{\mathbb{A}}, \ldots, u_n^{\mathbb{A}} \rangle^\sigma$ is $(\sigma^\uparrow, \sigma^\uparrow)$-continuous as well, whence by $\gamma^\uparrow \subseteq \sigma^\uparrow$ it is $(\sigma^\uparrow, \gamma^\uparrow)$-continuous. Thus the $(\gamma^\uparrow, \gamma^\uparrow)$-continuity of $s^{\mathbb{A}^\sigma}$ *matches* with the $(\sigma^\uparrow, \gamma^\uparrow)$-continuity of $\langle u_1^{\mathbb{A}}, \ldots, u_n^{\mathbb{A}} \rangle^\sigma$. Hence, we may apply Proposition 118(2), and find that $t^{\mathbb{A}^\sigma} = (s^{\mathbb{A}})^\sigma \circ \langle u_1^{\mathbb{A}}, \ldots, u_n^{\mathbb{A}} \rangle^\sigma \leq (s^{\mathbb{A}} \circ \langle u_1^{\mathbb{A}}, \ldots, u_n^{\mathbb{A}} \rangle)^\sigma = (t^{\mathbb{A}})^\sigma$, as desired. ❑

As a sample application, we show how Sahlqvist canonicity is an easy consequence of the previous theorem.

COROLLARY 124. *Sahlqvist equations are canonical over the class of all Boolean algebras with τ-operators.*

Proof. First we treat inequalities of the form $\varphi(\beta_1, \ldots, \beta_n) \leq \psi$, where φ only uses $\wedge$, $\vee$ and modalities, all β_i are boxed atoms, and ψ is positive. But then it is immediate by the previous proposition that $\varphi(\beta_1, \ldots, \beta_n)$ is stable, while ψ is expanding. Hence the result follows from Proposition 120.

Now consider an arbitrary Sahlqvist inequality. Without loss of generality we may assume that it is in fact an equation of the form

$$\varphi(\beta_1, \ldots, \beta_n, \neg\psi_1, \ldots, \neg\psi_k) \approx \bot, \tag{34}$$

where φ and the β's are as before, while all ψ_j are positive formulas. It is easy to see that this equation is equivalent to the quasi-equation

$$\left(\mathop{\&}_{1 \leq i \leq n} x_i \leq \neg\psi_i \right) \;\Rightarrow\; \varphi(\beta_1, \ldots, \beta_n, x_1, \ldots, x_k) \approx \bot,$$

which in its turn is equivalent to

$$\left(\mathop{\&}_{1 \leq i \leq n} x_i \wedge \psi_i \approx \bot \right) \;\Rightarrow\; \varphi(\beta_1, \ldots, \beta_n, x_1, \ldots, x_k) \approx \bot. \tag{35}$$

Now suppose that we *add* a diamond E to the language, and interpret this diamond as the global modality on every algebra (see section 8.2). Then clearly the quasi-equation (35) is equivalent to the formula

$$\varphi(\beta_1, \ldots, \beta_n, x_1, \ldots, x_k) \preceq \bigvee_{1 \leq i \leq n} \mathsf{E}(x_i \wedge \psi_i). \tag{36}$$

(Note that this reduction of a quasi-equation to an equivalent equation is a specific example of Proposition 138.)

The result then follows by the observation that (36) is a Sahlqvist inequality of the kind already treated, together with the fact that the canonical extension of the global modality is again the global modality (see Remark 110). ❑

7.6 Further remarks

The ideas described in this section allow for variations and generalizations in at least two directions.

To start with, the algebraic approach has already been put to work for a far wider class of structures than just Boolean algebras with operators. In particular, nothing in the theory crucially depends on the *Boolean* nature of the underlying order of the algebras. The notion of a canonical extension, with all the results in section 7.2 pertaining to them, has been extended to (first distributive and then) arbitrary lattices, with work on partial orders under way.

Furthermore, the restriction to monotone operations is not necessary either; *arbitrary* maps between lattices can be extended to maps between their canonical extensions. First suppose that we are dealing with a dense set X' in a topology $\langle X, \rho\rangle$, and let $f : X' \to C$ be a map from X' to the carrier C of a complete lattice $\mathbb{C}$. Then define

$$\begin{array}{rcl} f^{\sigma}(x) & := & \bigvee\{\bigwedge f[U \cap X'] \mid x \in U \in \rho\}, \\ f^{\pi}(x) & := & \bigwedge\{\bigvee f[U \cap X'] \mid x \in U \in \rho\}. \end{array} \tag{37}$$

In order to apply this definition for the canonical extension of a map f between two lattices $\mathbb{L}$ and $\mathbb{M}$, note that (just like in the case for Boolean algebras, see Remark 115) the carrier L of $\mathbb{L}$ forms a dense subset of the σ-topology over the carrier $\mathbb{L}^{\sigma}$. Also observe that f^{σ} and f^{π} are *extensions* of f because all elements of L are *isolated* points of f, and that for *monotone* f, (37) agrees with (32).

Finally, it is not just the definitions that translate to the more general setting of lattice expansions (that is, lattices with additional operations), the same holds for the theory. To mention just one example: one may prove that any equation $s \approx t$ is canonical provided that all the primitive symbols (including the join operation $\wedge$) occurring in s and t are interpreted as operators. Details can be found in for instance Gehrke & Harding [28].

The second generalization that we want to mention involves other ways of completing lattices and lattice expansions, such as the *MacNeille completion*, which generalizes Dedekind's construction of the reals from the rationals to arbitrary partial orders. For a characterization in the style of this section, one may start by proving that any lattice $\mathbb{L}$ has a (modulo isomorphism) unique completion $\mathbb{L}^{\mu}$, its *MacNeille completion*, in which $\mathbb{L}$ is both join- and meet dense. This way of extending lattices is obviously similar to that of the canonical extension, but a substantial difference is that the MacNeille completion agrees with the original lattices on *all* meets and joins, whereas the canonical extension only agrees on the *finite* ones.

In any case, it follows from join- and meet density, that any map between two lattices can be extended to a map between their MacNeille completions, in two ways. In the case of a *monotone* operation f between two lattices $\mathbb{L}$ and $\mathbb{M}$, we define the *lower extension* $\check{f}$ and the *upper extension* $\hat{f}$ by

$$\begin{array}{rcl} \check{f}(x) & := & \bigvee\{f(a) \mid x \geq a \in L\} \\ \hat{f}(x) & := & \bigwedge\{f(a) \mid x \leq a \in L\} \end{array}$$

Clearly then, almost all questions concerning canonical extensions have an obvious counterpart for MacNeille completions. Generally speaking, MacNeille completions are less

well-behaved than canonical extensions; for instances, unary operators (diamonds) are no longer smooth, and the variety of modal algebras is not closed under taking lower MacNeille completions. Probably for this reason, Monk [85] introduced the notion of the MacNeille completion of a BAO only for Boolean algebras with *complete* operators. On the other hand, in case the primitive operations are *residuated* (see Proposition 129), the situation improves; for instance, Givant & Venema [36] show that the validity of all Sahlqvist equations is preserved under taking MacNeille completions of *tense* algebras. As a final remark, there are interesting connections between the MacNeille completion and the canonical extension of a lattice expansion: for instance, Gehrke, Harding & Venema [29] prove that the canonical extension of lattice expansion $\mathbb{A}$ can be embedded in the MacNeille completion of some ultrapower of $\mathbb{A}$. As a consequence, every variety of lattice expansions that is closed under taking MacNeille completions, is also canonical in the sense of canonical extensions.

8 SPECIAL ALGEBRAIC TOPICS

In this final section on algebra we discuss the algebraic perspective on two further issues in modal logic.

8.1 Tense logic

Our first example concerns *tense logic*; as its name already indicates, this branch of modal logic originates in the formal semantics of natural language, cf. Chapter 19 of this volume.

DEFINITION 125. The modal similarity type ϑ of *tense logic* is fixed by its two diamonds, $\Diamond_F$ and $\Diamond_P$.

The letters $\Diamond_F$ and $\Diamond_P$ are mnemonic of **f**uture and **p**ast, respectively. This already indicates that the standard interpretation of this language is in frames representing a flow of time, such that $\Diamond_F$ obtains the meaning 'sometime in the future', and dually $\Diamond_P$ means 'sometime in the past'. Tense logic thus forms a rather simple example of *temporal* logic, cf. Chapter 11 of this volume. Here we abstract from the temporal interpretations of tense logics; what is then left is that in the intended frames for this language, the two diamonds of the language are interpreted along the two directions of a *single* binary relation.

DEFINITION 126. A ϑ-frame $\mathbb{S} = \langle S, R_F, R_P\rangle$ is called *bidirectional* if R_F and R_P are each other's converse.

This definition explains why a ϑ-frame is often represented simply as the pair $\langle S, R_F\rangle$. Turning to logic, we define the following.

DEFINITION 127. A modal ϑ-logic L is a *tense logic* if both formulas $p \to \Box_F \Diamond_P p$ and $p \to \Box_P \Diamond_F p$ are theorems of L; the minimal tense logic is denoted as $\mathbf{K}_t$. Algebraically, a *tense algebra* is a Boolean algebra expanded with monotone ϑ-operations satisfying the corresponding equations $x \preceq \Box_F \Diamond_P x$ and $x \preceq \Box_P \Diamond_F x$.

It is easy to see that $\mathbb{S}^+$ is a tense algebra if and only if $\mathbb{S}$ is a bidirectional frame. In the other direction, it is not a priori clear whether we can extract a useful frame from

an arbitrary tense algebra: First we must show that tense algebras are Boolean algebras with operators. In fact, already Jónsson & Tarski [70] show something better.

THEOREM 128. *Let* $\mathbb{A} = \mathbb{A} = \langle A, \top, \bot, -, \wedge, \vee, \Diamond_F, \Diamond_P \rangle$ *be a tense algebra. Then*

1. *the operations* $\Diamond_F$ *and* $\Diamond_P$ *are complete operators;*
2. *the structure* $\mathbb{A}_\bullet$ *is a bidirectional frame, and the algebra* $\mathbb{A}^\sigma$ *is again a tense algebra.*

Proof. For part 1 of the Theorem, let $a \in A$ be the least upper bound of some subset X of A. Then by monotonicity, $\Diamond_F a$ is an upper bound of the set $\Diamond_F[X]$. Now suppose that b is also an upper bound of this set, that is, $\Diamond_F x \leq b$ for all $x \in X$. From this it follows, for each $x \in X$, that $x \leq \Box_P \Diamond_F x \leq \Box_P b$ (here we use monotonicity of $\Box_P$, which is easily proven). Thus we see that $a \leq \Box_P b$ by our assumption on a. But then by monotonicity of $\Diamond_F$ we obtain that $\Diamond_F a \leq \Diamond_F \Box_P b \leq b$. This proves that $\Diamond_F a$ is in fact the least upper bound of the set $\Diamond_F[X]$.

Concerning the second part of the theorem, that $\mathbb{A}^\sigma$ is a tense algebra is a special of the Sahlqvist Canonicity Theorem 93; the bidirectionality of $\mathbb{A}_\bullet$ is then immediate since $\mathbb{A}^\sigma = (\mathbb{A}_\bullet)^+$. ❑

There is a lot more to say about the complete additivity of the diamonds in tense algebras. To start with, the definition of tense algebras can be reformulated using either of the algebraically more familiar notions of *conjugation* or *residuation.*

PROPOSITION 129. *Let* $\mathbb{A} = \langle A, \top, \bot, -, \wedge, \vee, \Diamond_F, \Diamond_P \rangle$ *be a monotone* ϑ*-expanded Boolean algebra. Then the following are equivalent:*

1. $\mathbb{A}$ *is a tense algebra,*
2. $\Diamond_F$ *and* $\Diamond_P$ *are conjugated operations, that is, they satisfy the following:*
$$\mathbb{A} \models \forall xy\, (x \wedge \Diamond_F y \approx \bot \Leftrightarrow y \wedge \Diamond_P x \approx \bot), \tag{38}$$
3. $\Diamond_F$ *and* $\Box_P$ *form a residual pair, that is,*
$$\mathbb{A} \models \forall xy\, (\Diamond_F x \preceq y \Leftrightarrow x \preceq \Box_P y). \tag{39}$$

This connection with residuation shows that from a general mathematical perspective, tense logic is not just *any* bimodal logic: It provides the modal logic manifestation of the fundamental category theoretic concept of *adjoint functors.* Theorem 128(1) is thus a rather special case of the category theoretic fact that left adjoint functors preserve all (existing) colimits.

Another nice property of tense logic that should be mentioned here is that somehow, tense algebras are *richer* than ordinary Boolean algebras with operators. For instance, consider an atomic modal algebra $\mathbb{A}$, and suppose that $\mathbb{A}$ satisfies some Sahlqvist equation η. Then it is *not* guaranteed that the atom structure $\mathbb{A}_\bullet$ (see Definition 41) satisfies the first-order correspondent c_η of η, not even if the diamond of $\mathbb{A}$ is completely additive. However, in case $\mathbb{A}$ is a *tense* algebra, it contains sufficient information to enforce this.

THEOREM 130. *Let* $\mathbb{A}$ *be an* atomic *tense algebra. Then for every Sahlqvist equation* η*:* $\mathbb{A} \models \eta$ *iff* $\mathbb{A}_+ \models c_\eta$ *iff* $(\mathbb{A}_\bullet)^+ \models \eta$.

Proof. Clearly, the equivalence of the last two statements follows from Sahlqvist correspondence theory. For the implication from right to left, it suffices to observe that $\mathbb{A}$ is a subalgebra of $(\mathbb{A}_\bullet)^+$ because of the complete additivity of the operators. This follows from (12) in the proof of Proposition 42.

The remaining implication is a special case of the preservation of Sahlqvist equations under taking (lower) MacNeille completions of tense algebras, see the end of section 7 for some discussion, and Givant & Venema [36] for proofs. ❑

Finally, tense algebras play a role in other part of universal algebra as well. For instance, any *lattice* can be represented as the sublattice of a tense algebra that has the solution set of the equation $x \approx \Box_P \Diamond_F x$ as its carrier. This idea basically goes back to Birkhoff [12]; for more details, the reader is referred to Harding [56].

Nevertheless, despite their rather special characteristics, just like all bimodal logics, tense logics can be *simulated* by monomodal ones; for details we refer to Chapter 8 of this volume.

8.2 Global modality & discriminator varieties

Recent years have witnessed an increasing interest in formalisms that enhance the expressive power of standard modal languages, see for instance Chapter 14 of this volume. In such a pursuit, one naturally arrives at the *global* or *universal* modality E which has the global relation $S \times S$ of a frame $\mathbb{S}$ as its (intended) accessibility relation, see Goranko & Passy [48]. But also, a large number of standard logics come with an intended semantics in which the global relation interprets some more complex term of the language: as an example we mention the compound modality $\Diamond_F \Diamond_P$ in the tense logic over any linear flow of time.

DEFINITION 131. Algebraically, we define the *global modality* or *unary discriminator* over a Boolean algebra (with operators) $\mathbb{B}$ as the function given by

$$b \mapsto \begin{cases} \bot & \text{if } b = \bot, \\ \top & \text{if } b > \bot. \end{cases}$$

The term $\gamma(x)$ is called a *global modality* or *unary discriminator term* over an expanded Boolean algebra $\mathbb{A}$ if it is interpreted as the global modality on $\mathbb{A}$.

This notion can be seen as the BAO manifestation of the well-known algebraic concept of a *discriminator*, see Jipsen [67] for a first explicit discussion of the connections.

DEFINITION 132. We call a ternary term d a *discriminator term* over an algebra $\mathbb{A}$ if it is interpreted as the discriminator function on A, that is, if $d^{\mathbb{A}}(a,b,c) = a$ if $a \neq b$, and $d^{\mathbb{A}}(a,b,c) = c$ if $a = b$. Any variety V generated by a class of algebras with a common discriminator term, is called a *discriminator* variety.

PROPOSITION 133. *Let $\mathbb{A}$ be a τ-expanded Boolean algebra.*

1. *If γ is a global modality for $\mathbb{A}$, then the term $(\gamma(\neg(x \leftrightarrow y)) \wedge x) \vee (\gamma(\neg(x \leftrightarrow y)) \wedge z)$ is a discriminator term for $\mathbb{A}$.*

2. *If $d(x,y,z)$ is a discriminator term for $\mathbb{A}$, then the term $\neg d(\bot, x, \top)$ is a global modality for $\mathbb{A}$.*

Before going into further detail of the connection with the global modality, let us, for future reference, list some of the many nice properties that discriminator varieties have.

THEOREM 134. *Let* K *be a class of algebras with a discriminator term* d*. Then*

1. *all algebras in* K *are simple;*
2. $\mathsf{Var}(\mathsf{K})$ *is congruence-distributive and congruence-permutable;*
3. *all subdirectly irreducible algebras in* $\mathsf{Var}(\mathsf{K})$ *are simple, and vice versa;*
4. $\mathsf{Var}(\mathsf{K})$ *is semi-simple; that is, every algebra in* $\mathsf{Var}(\mathsf{K})$ *is a subdirect product of simple algebras.*
5. d *is a discriminator term for every simple algebra in* $\mathsf{Var}(\mathsf{K})$*.*

Proof. For the first statement of the theorem, define the term

$$s(x, y, u, v) := d(d(x, y, u), d(x, y, v), v).$$

It is easy to see that s is a so-called *switching term* for K; that is, for every $\mathbb{A}$ in K, and for all a, b, c and d in $\mathbb{A}$:

$$s^{\mathbb{A}}(a, b, c, d) = \begin{cases} c & \text{if } a = b, \\ d & \text{if } a \neq b. \end{cases}$$

Now let $\Theta \neq \Delta_A$ be a congruence of $\mathbb{A}$; then there are two elements $a \neq b$ with $(a, b) \in \Theta$. But then we find $(c, d) = (s^{\mathbb{A}}(a, a, c, d), s^{\mathbb{A}}(a, b, c, d)) \in \Theta$ for every c and d in Θ. In other words, such a Θ must be the trivial congruence $A \times A$. But this clearly means that $\mathbb{A}$ is simple. Details of the proof of the second statement, which is similar to that of Theorem 25, are left to the reader.

For the third part of the theorem, it is not hard to verify that d is a discriminator term for $\mathsf{SPu}(\mathsf{K})$ as well, whence $\mathsf{SPu}(\mathsf{K})$ consists of simple algebras by part (1). So by definition of simplicity, we find that $\mathsf{HSPu}(\mathsf{K}) = \mathsf{SPu}(\mathsf{K})$; hence, all algebras in $\mathsf{HSPu}(\mathsf{K})$ are simple. However, by part 2 we may apply Jónsson's Lemma, which states that all s.i. members of $\mathsf{Var}(\mathsf{K})$ belong to $\mathsf{HSPu}(\mathsf{K})$. Thus every s.i. algebra in $\mathsf{Var}(\mathsf{K})$ is simple.

Part (4) is immediate from part (3) by Birkhoff's subdirect indecomposability theorem, while the final statement follows from the fact that every simple algebra belongs to $\mathsf{SPu}(\mathsf{K})$, and thus shares the discriminator term of K. ❑

In particular, since the notions of simplicity and subdirect irreducibility coincide in a discriminator variety, its subvarieties are completely determined by its simple members. Let us now see how these issues are axiomatized in normal modal logics.

DEFINITION 135. A τ-formula $\gamma(x)$ is a *global modality* for a normal modal τ-logic L if the formulas Γ

- $\nabla(x_1, \ldots, x_n) \rightarrow \gamma(x_i)$ for every $\nabla \in \tau$, and every $i \in \{1, \ldots, n\}$;
- $x \rightarrow \gamma(x)$, $\gamma(\gamma(x)) \rightarrow \gamma(x)$ and $\gamma(\neg\gamma(\neg x)) \rightarrow x$;

are theorems of L.

That is, L defines a global modality iff there is a term $\gamma(x)$ that satisfies the **S5** axioms, plus the inclusion axiom $\nabla^i x \to \gamma(x)$ for every induced diamond ∇^i. It is not hard to derive that such an axiomatically defined global modality $\gamma(x)$ also has $\gamma(\neg\gamma(x)) \to \neg\gamma(x)$, and $\vdash_L \blacklozenge x \to \gamma(x)$ for all compound diamonds $\blacklozenge$.

The terminology of Definition 135 is justified by the following Proposition, which is essentially taken from Jipsen [67].

PROPOSITION 136. *Let L be a normal modal τ-logic, and $\gamma(x)$ a τ-formula. Then $\gamma(x)$ is a global modality for L if and only if $\mathsf{BAO}_\tau(L) = \mathsf{Var}(\mathsf{K})$ for some class K of algebras sharing γ as a global modality.*

Proof. The direction from right to left is immediate by the fact that any unary discriminator term satisfies all the formulas listed in Definition 135.

For the other direction, by Theorem 34 it suffices to show that γ is a unary discriminator term on subdirectly irreducible algebras in $\mathsf{BAO}_\tau(L)$. In order to prove this, suppose for contradiction that $\mathbb{A}$ has a radical element ρ, while $\gamma^{\mathbb{A}}$ is not the global modality on $\mathbb{A}$. That is, some $a \in A$ satisfies $a \neq \bot$ while $\gamma^{\mathbb{A}}(a) \neq \top$, whence $-\gamma^{\mathbb{A}}(a) \neq \bot$. Since ρ is radical in $\mathbb{A}$ there are compound diamonds $\blacklozenge_1$ and $\blacklozenge_2$ such that $\rho \leq \blacklozenge_1 a$ and $\rho \leq \blacklozenge_2 -\gamma^{\mathbb{A}}(a)$. However, from $\rho \leq \blacklozenge_1 a$ we obtain $\rho \leq \gamma^{\mathbb{A}}(a)$, while from $\rho \leq \blacklozenge_2 -\gamma^{\mathbb{A}}(a)$ we may infer that $\rho \leq \gamma^{\mathbb{A}}(-\gamma^{\mathbb{A}}(a)) \leq -\gamma^{\mathbb{A}}(a)$. This contradicts the fact that $\rho > \bot$, and so we may conclude that γ is the global modality on $\mathbb{A}$. ❑

A very useful property of discriminators is that they allow the effective replacement of universal sentences with equations. In the case of BAOs, this works out as follows.

DEFINITION 137. Suppose that $\gamma(x)$ is a global modality term for K. Inductively we define a function λ mapping quantifier-free formulas (in the first order language of BAOs) to τ-terms:

$$\begin{array}{rcl} s \approx t & \mapsto & (s \wedge \neg t) \vee (\neg s \wedge t), \\ \sim P & \mapsto & \neg\gamma(\lambda_P), \\ P \mathbin{\&} Q & \mapsto & \lambda_P \vee \lambda_Q. \end{array}$$

THEOREM 138. *Let K be a class of Boolean algebras with τ-operators with a discriminator term γ. Then any universal formula P is equivalent over K to the equation $\lambda_{P'} \approx \bot$, where P' is the quantifier-free part of P.*

Proof. A straightforward induction shows that for any algebra $\mathbb{A}$ in K, any assignment α on $\mathbb{A}$ and any quantifier-free formula P it holds that

$$\mathbb{A} \models_\alpha P \text{ iff } \mathbb{A} \models_\alpha \lambda_P \approx \bot.$$

From this, the statement of the theorem is immediate. ❑

Working with discriminator classes has many advantages. For instances, if K is a discriminator class, then we may generate $\mathsf{Var}(\mathsf{K})$ from K just by taking products and subalgebras (that is, homomorphic images are not needed). The result in this generality is due to Givant [35].

THEOREM 139. *Let K be a class of Boolean algebras with a common global modality term $\gamma(x)$.*

1. *If* $\mathsf{Pu}(\mathsf{K}) \subseteq \mathsf{S}(\mathsf{K})$*, then* $\mathsf{SP}(\mathsf{K})$ *is a variety and* $\mathsf{S}(\mathsf{K})$ *is the universal class of simple algebras in* $\mathsf{SP}(\mathsf{K})$.

2. *If* K *is axiomatized by a set* Φ *of universal formulas, then* $\mathsf{SP}(\mathsf{K})$ *is axiomatized by the set* $\{\lambda_P \approx \bot \mid P \in \Phi\}$*, together with the set* Γ *of Definition 135.*

Proof. Assume that $\mathsf{Pu}(\mathsf{K}) \subseteq \mathsf{S}(\mathsf{K})$, then it is easy to see that the class $\mathsf{S}(\mathsf{K})$ is closed under taking ultraproducts and subalgebras. It then follows by standard universal algebra, see [17, Theorem 2.20], that $\mathsf{S}(\mathsf{K})$ is a universal class, that is, an elementary class axiomatized by universal formulas.

By assumption, the algebras in K have a common discriminator term, and, hence, we find, reasoning as in the proof of Theorem 134(3), that $\mathsf{SirVar}(\mathsf{K}) = \mathsf{SPu}(\mathsf{K})$, where $\mathsf{SirVar}(\mathsf{K})$ denotes the class of s.i. members in $\mathsf{Var}(\mathsf{K})$. Thus by the assumption we find that $\mathsf{SirVar}(\mathsf{K}) = \mathsf{S}(\mathsf{K})$ and therefore, $\mathsf{S}(\mathsf{K})$ is the class of simple algebras in $\mathsf{Var}(\mathsf{K})$, since the notions of simplicity and subdirect irreducibility coincide. Finally then, by Birkhoff's and Jónsson's theorems, the variety $\mathsf{Var}(\mathsf{K})$ is the class of subdirect products of algebras in $\mathsf{HSPu}(\mathsf{K}) = \mathsf{S}(\mathsf{K})$; a straightforward calculation then will show that $\mathsf{Var}(\mathsf{K}) = \mathsf{SP}(\mathsf{K})$.

Part two of the theorem is a straightforward consequence of Proposition 136 and Theorem 138. ❑

Finally, for more information on the global modality, the reader is referred to Chapter 8 of this volume.

9 COALGEBRAS: AN INTRODUCTION

This section forms a brief introduction to the field of *Coalgebra.* While certain kinds of coalgebras had already been studied in the sixties, the field really took off after it was realized that coalgebra can be conceived as a general and uniform theory of dynamic systems, taken in a broad sense.

Many structures in mathematics and theoretical computer science can naturally be represented as coalgebras. Probably the first example was provided by Aczel [2], who models transition systems and non-well-founded sets as coalgebras. On the basis of Aczel's work, Barwise & Moss [11] discuss a wide range of phenomena involving the notions of circularity and self-reference, with applications ranging from theoretical economics to the semantics of natural language. A second paradigmatic specimen of coalgebras in computer science is given by (deterministic) automata, see Rutten [96]. Further important examples include the representation of infinite data structures, and the formal modeling of objects and classes in object oriented programming, see Reichel [92] or Jacobs [61]. But for modal logicians, it will be Kripke frames and models that provide the prime examples of coalgebras; this link goes back to at least Abramsky [1]. In fact, the model theory of modal logic is coalgebraic in nature, so modal logicians entering the field will have much the same experience as group theorists learning about universal algebra, in that they will recognize many familiar notions and results, lifted to a higher level of generality and abstraction.

For readers that want to learn more about coalgebras, the literature harbors some well written introductions and surveys (although at the time of writing there is no text book or monograph available). We refer the reader to Jacobs & Rutten [65] for a very accessible

introduction, and to Rutten [97] or Gumm [50] for comprehensive surveys. Ihringer [60] has an appendix on coalgebras by Gumm. For more details on the connection between coalgebra and modal logic, the reader may consult Kurz [75] or Pattinson [90].

What then are coalgebras? The most concrete, state-based specimens, called *systems*, simply consist of a set S endowed with some kind of transition, formally modeled as some map σ from S to another set ΩS. Here Ω is some functor constituting the *type* or *signature* of the coalgebra at stake. The transition map provides some kind of structure on S, but whereas *algebraic* operations are ways to *construct* complex objects out of simple ones, coalgebraic operations, going *out of* the carrier set, should be seen as ways to *unfold* or *observe* objects. This explains the central role of the notion of *behavior* in the theory of coalgebras.

More generally, given an endofunctor Ω on some base category C, an Ω-*coalgebra* is a pair $\mathbb{C} = \langle C, \gamma \rangle$, with C an arbitrary object in C, and γ a C-arrow from C to ΩC. The full functorial power of Ω comes in when we turn Ω-coalgebras into a category $\mathsf{Coalg}(\Omega)$ by introducing morphisms: A homomorphism from $\langle C, \gamma \rangle$ to $\langle C', \gamma' \rangle$ is an arrow $f : C \to C'$ such that $\gamma' \circ f = (\Omega f) \circ \gamma$. This set-up enables the canonical definition of two notions of equivalence between coalgebras, namely, bisimulation and behavioral equivalence. As we will see as well, the definitions make the concept of a coalgebra very similar to that of an algebra. However, if one makes this connection mathematically precise, it turns out that coalgebras over the base category C are *dual* to algebras over the *opposite* category C^{op}. This explains not only the name 'coalgebra', but, as we will see, also many of the peculiarities of *universal coalgebra*, that is, the general coalgebraic theory of systems.

Given the nature of coalgebra as a very general model of state-based dynamics, there is a natural place for *modal logic* as a formalism for reasoning about behavior. It was Moss [11, 86] who realized that one may generalize the concept of modal logic from Kripke frames and models to coalgebras over arbitrary set functors. Over subsequent years, the development and study of modal languages for the specification of properties of coalgebras has been actively pursued and studied by various authors, including Jacobs [62, 64], Kurz [77, 76], Pattinson [88, 89], and Rößiger [95]. In fact, as we will see, the link between modal logics and coalgebra is so tight, that one may even claim that modal logic is the natural logic for coalgebras — just like equational logic is that for algebra.

We now turn to the technical development of the topic, starting with the definition of a *coalgebra*.

DEFINITION 140. Given an endofunctor Ω on a category C, an Ω-*coalgebra* is a pair $\mathbb{A} = (A, \alpha)$, where A is an object of C called the *carrier* of $\mathbb{A}$, and $\alpha : A \to \Omega A$ is an arrow in C, called the *transition map* of $\mathbb{A}$. In case Ω is an endofunctor on Set, Ω-coalgebras may also be called Ω-*systems*; a *pointed* Ω-*system* is a triple $\langle A, \alpha, a \rangle$ such that $\langle A, \alpha \rangle$ is an Ω-system, and a is a state in $\mathbb{A}$, that is, an element of A.

As we mentioned already, the action of the functor Ω on the *arrows* of the category C will be needed when we introduce, in Definition 148 below, homomorphisms between Ω-coalgebras. First we consider some examples of systems.

EXAMPLE 141. Probably the simplest example of a system is that of an C-colored set, that is, a pair $\langle S, \gamma : S \to C \rangle$. No matter where we start, this system can only display the color of the current state, and halt after doing so.

A slightly more interesting example is provided by a black box machine which may be prompted to display a value, or color, from C, and to move on to a next state. These

states are internal to the machine, that is, invisible to an outside observer. Such a machine can abstractly be modeled as a coalgebra $\mu : M \to C \times M$, with $\pi_0(\mu(s)) \in C$ denoting the current value of the machine, and $\pi_1(\mu(s)) \in M$ representing the machine's next internal state. (Here $\pi_0 : C \times M \to C$ and $\pi_1 : C \times M \to M$ are the projection functions.)

EXAMPLE 142. For our second example, we turn to automata theory. Recall that *deterministic automata* are usually modeled as quintuples $\mathbb{A} = \langle A, a_I, C, \delta, F\rangle$ such that A is the state space of the automaton $\mathbb{A}$, $a_I \in A$ is its initial state, C its *alphabet*, $\delta : A \times C \to A$ its *transition function* and finally, $F \subseteq A$ its collection of *accepting states*.

Now observe that we may represent F by its characteristic map $\chi_F : A \to 2$ (with 2 denoting the set $\{0, 1\}$) which maps $a \in A$ to 1 if $a \in F$, and to 0 if $a \notin F$. Furthermore, we can and will view δ as a map from $A \to A^C$, where A^C denotes the collection of maps from C to A. Thus we see that we may represent a deterministic automaton over the alphabet C as a pointed system over the functor $S \mapsto 2 \times S^C$.

EXAMPLE 143. Our third example provides the crown witness when it comes to the connection between coalgebra and modal logics: We will now see that *frames* and *models* are in fact coalgebras in disguise. The crucial observation is here that a binary relation $R \subseteq S \times S$ can be represented by the function $R[\cdot] : S \to \mathcal{P}(S)$ mapping a point s to the collection $R[s]$ of its successors. Thus frames for the basic modal similarity type correspond to coalgebras over the covariant power set functor $\mathcal{P}$. (This functor maps a set S to its power set $\mathcal{P}(S)$ and a function $f : S \to S'$ to the image map $\mathcal{P}f$ given by $(\mathcal{P}f)(X) := f[X](= \{f(x) \mid x \in X\})$.)

Similarly, a ternary relation $T \subseteq S^3$ can be modeled as the function $T[\cdot] : S \to \mathcal{P}(S^2)$ given by $T[s] = \{(t_1, t_2) \in S^2 \mid Tst_1t_2\}$. Thus for any modal similarity type τ, we can represent τ-frames as coalgebras for the functor $S \mapsto \prod_{\nabla\in\tau} \mathcal{P}(S^{ar(\nabla)})$. Also note that *image finite frames*, that is, frames in which $R[s]$ is a finite set for all points s, correspond to coalgebras over the *finitary* power set functor $\mathcal{P}_\omega$.

Concerning models, in this section we let Prop denote the set of propositional variables. It is easy to see that a valuation $V : \mathsf{Prop} \to \mathcal{P}(S)$ on a frame $\mathbb{S} = \langle S, R\rangle$ could equivalently have been defined as a $\mathcal{P}(\mathsf{Prop})$-coloring of S, that is, as the map sending a state s to the collection $V^{-1}[s] = \{p \in \mathsf{Prop} \mid s \in V(p)\}$ of proposition letters holding at s. Thus models for the basic modal similarity type can be identified with coalgebras of the functor Ω given by $X \mapsto \mathcal{P}(\mathsf{Prop}) \times \mathcal{P}(X)$.

EXAMPLE 144. For our last example, let $\breve{\mathcal{P}}$ denote the *contravariant* power set functor. This functor agrees with the covariant power set functor on objects, while on arrows $\breve{\mathcal{P}}$ takes *inverse* images. That is, for $f : A \to A'$, the function $\breve{\mathcal{P}}f : \mathcal{P}A' \to \mathcal{P}A$ is given by $(\breve{\mathcal{P}}f)(X') := f^{-1}[X'](= \{x \in A \mid f(x) \in X'\})$. Note that $\breve{\mathcal{P}}$ is not a functor from Set to Set, and thus does not produce coalgebras. Its composition with itself, however, *is* an endofunctor on Set, so that we may consider $\breve{\mathcal{P}} \circ \breve{\mathcal{P}}$-coalgebras. Because the transition function σ of such a coalgebra $\langle S, \sigma\rangle$ is a function $\sigma : S \to \mathcal{P}\mathcal{P}S$, the structure $\langle S, \sigma\rangle$ may also be seen as a *neighborhood frame*, as discussed in Chapter 1 of this volume.

Some variants of the functor $\breve{\mathcal{P}} \circ \breve{\mathcal{P}}$ are of interest as well — we discuss the examples $\mathcal{U}_{\breve{\mathcal{P}}}$ and $\mathcal{F}_{\breve{\mathcal{P}}}$. Recall that $\breve{\mathcal{P}} \circ \breve{\mathcal{P}}(S) = \mathcal{P}\mathcal{P}(S)$ is the set of *all* collections of subsets of S. $\mathcal{U}_{\breve{\mathcal{P}}}(S)$ denotes the set of all *upward closed* collections of subsets of S, while $\mathcal{F}_{\breve{\mathcal{P}}}(S)$ denotes the set of all *filters* of S. On arrows, these functors coincide with $\breve{\mathcal{P}} \circ \breve{\mathcal{P}}$; more precisely, for $f : S \to S'$, we set $\mathcal{U}_{\breve{\mathcal{P}}}f$ and $\mathcal{F}_{\breve{\mathcal{P}}}f$ as the restrictions of $(\breve{\mathcal{P}} \circ \breve{\mathcal{P}})f$ to $\mathcal{U}_{\breve{\mathcal{P}}}S$

and $\mathcal{F}_{\breve{\mathcal{P}}}S$, respectively.

It is not hard to show that $\mathcal{U}_{\breve{\mathcal{P}}}$ and $\mathcal{F}_{\breve{\mathcal{P}}}$ are indeed functors $\mathsf{Set} \to \mathsf{Set}$. The reader may in fact be familiar with (some) coalgebras for these functors. It can easily be verified that the $\mathcal{U}_{\breve{\mathcal{P}}}$-coalgebras correspond exactly to the *monotonic neighborhood frames* that were mentioned in Chapter 1 as the superset closed neighborhood frames. Prime examples of $\mathcal{F}_{\breve{\mathcal{P}}}$-coalgebras are the topological spaces (that were also mentioned in Chapter 1, be it rather implicitly under the name of *topological semantics*). To see this, represent the topology σ on the set S by the function mapping a point $s \in S$ to the collection $\{U \in \sigma \mid s \in U\}$ of its neighborhoods.

EXAMPLE 145. For each set functor Ω, the empty set $\varnothing$, with the unique map from $\varnothing$ to $\Omega\varnothing$, provides an Ω-coalgebra.

The functors mentioned in the Examples 141, 142 and 143, are examples of so-called *Kripke polynomial functors* which share some pleasant properties as we will see further on.

DEFINITION 146. The collection of *polynomial functors* is inductively defined as follows:

$$K \;::=\; \mathcal{I} \mid C \mid K_0 + K_1 \mid K_0 \times K_1 \mid K^D. \tag{40}$$

Here $\mathcal{I}$ denotes the identity functor on the category Set; C the constant functor $X \mapsto C$; $K_0 + K_1$ the coproduct functor $X \mapsto K_0(X) + K_1(X)$; $K_0 \times K_1$ the product functor; and K^D denotes the exponent functor $X \mapsto K(X)^D$.

Similarly, the collection of *Kripke polynomial functors* is given by

$$K \;::=\; \mathcal{I} \mid C \mid K_0 + K_1 \mid K_0 \times K_1 \mid K^D \mid \mathcal{P}K, \tag{41}$$

where $\mathcal{P}K$ is the composition of K with the power set functor $\mathcal{P}$. Replacing $\mathcal{P}$ with the *finite power set functor* $\mathcal{P}_\omega$, and demanding the exponent D in K^D to be finite, we obtain the collection of *finitary Kripke polynomial functors*.

In each of these cases, the set $Ing K$ of *ingredient functors* of a (Kripke) polynomial functor K is defined by an obvious induction, with clauses $Ing(\mathcal{I}) := \{\mathcal{I}\}$, $Ing(\mathcal{P}K) := \{\mathcal{P}K\} \cup Ing(K)$, etc.

With the notation of this definition, Example 141 provides examples of coalgebras for the functors C and $\mathcal{I} \times C$. Deterministic automata over the alphabet C are $2 \times \mathcal{I}^C$-coalgebras. Kripke frames are $\mathcal{P}\mathcal{I}$-coalgebras, and Kripke models are coalgebras for the functor $\mathcal{P}\mathsf{Prop} \times \mathcal{P}\mathcal{I}$. (Note that in the format (41), the power set functor as such is not a Kripke polynomial functor: It has to be represented as the functor $\mathcal{P}\mathcal{I}$. In the sequel, we will keep working with Kripke frames as $\mathcal{P}$-coalgebras, unless explicitly mentioned otherwise.)

After Set, the base category for coalgebras that carries most interest to modal logicians, is probably that of *Stone spaces*.

EXAMPLE 147. Recall from Remark 60 that a Stone space is pair $\mathbb{S} = (S, \sigma)$ such that σ is a compact Hausdorff space with a basis of clopens. Let Stone denote the category with Stone spaces as objects, and continuous maps as arrows. We will show that *descriptive general frames* can be viewed as Stone-coalgebras for the so-called *Vietoris functor* $\mathcal{V}$ — for details on this observation, which is due to Abramsky [1], see Kupke, Kurz & Venema [74].

This functor, which forms the topological counterpart of the power set functor, is defined as follows. Given a topological space $\mathbb{S} = \langle S, \sigma\rangle$, let $K(\mathbb{S})$ denote the collection of closed subsets of S, and let $\ni \subseteq K(\mathbb{S}) \times S$ denote the converse membership relation. Then (in accordance with our earlier notation), we define, for any subset $U \subseteq S$, the sets $\langle\ni\rangle U = \{F \in K(\mathbb{S}) \mid F \cap U \neq \varnothing\}$ and $[\ni]U = \{F \in K(\mathbb{S}) \mid F \subseteq U\}$. The topology on $K(\mathbb{S})$, generated by taking the collection $\{\langle\ni\rangle U, [\ni]U \mid U \in \sigma\}$ as a subbasis, is called the *Vietoris topology* of σ, and the resulting space, the *Vietoris space* $\mathcal{V}(\mathbb{S})$ associated with $\mathbb{S}$.

The Vietoris construction preserves several properties of topological spaces; in particular, if $\mathbb{S}$ is a Stone space, then so is $\mathcal{V}(\mathbb{S})$. Also, we may extend it to a functor, by defining, for a continuous map $f : \mathbb{S} \to \mathbb{S}'$, the function $\mathcal{V}f$ as the image map given by $(\mathcal{V}f)(X) := f[X]$. Here we omit the proof that $\mathcal{V}f$ is indeed an arrow in the category Stone, i.e., that it is a *continuous* map from $\mathcal{V}(\mathbb{S})$ to $\mathcal{V}(\mathbb{S}')$.

Now let $\mathbb{G} = \langle G, R, A\rangle$ be a descriptive general frame (cf. Definition 64), with associated Stone space σ_A. Recall from Remark 65 that the map $R[\cdot]$ mapping a point in G to the collection of its successors, is a function from G to $K(\langle G, \sigma_A\rangle)$. It is not too hard to prove that this is in fact a *continuous* map from $\langle G, \sigma_A\rangle$ to its Vietoris space. Thus we may represent $\mathbb{G}$ as the Stone coalgebra $\langle\langle G, \sigma_A\rangle, R[\cdot]\rangle$.

Obviously, coalgebras are not studied in isolation; the following definition provides a natural notion of a map between coalgebras that preserves the transition structure.

DEFINITION 148. Let $\mathbb{A} = \langle A, \alpha\rangle$ and $\mathbb{A}' = \langle A', \alpha'\rangle$ be two coalgebras for the functor $\Omega : \mathsf{C} \to \mathsf{C}$. Then a *homomorphism* from $\mathbb{A}$ to $\mathbb{A}'$ is an arrow $f : A \to A'$ for which the following diagram commutes:

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\alpha"'] & A' \arrow[d, "\alpha'"] \\ \Omega A \arrow[r, "\Omega f"] & \Omega A' \end{tikzcd}$$

EXAMPLE 149. The homomorphisms for $\mathcal{P}$-coalgebras coincide with the bounded morphisms between Kripke frames. To see this, let $\mathbb{S} = \langle S, R\rangle$ and $\mathbb{S}' = \langle S', R'\rangle$ be two frames (for the basic modal similarity type), and consider their respective coalgebraic representations $\langle S, \sigma\rangle$ and $\langle S', \sigma'\rangle$, as in Example 143.

Now consider a map $f : S \to S'$. It is straightforward to show that

$$\begin{array}{lll} f \text{ satisfies the forth condition} & \text{iff} & (\mathcal{P}f) \circ \sigma(s) \subseteq \sigma' \circ f(s) \text{ for all } s \in S, \\ f \text{ satisfies the back condition} & \text{iff} & (\mathcal{P}f) \circ \sigma(s) \supseteq \sigma' \circ f(s) \text{ for all } s \in S. \end{array}$$

This shows that f is a bounded morphism from $\mathbb{S}$ to $\mathbb{S}'$ if and only if it is a coalgebra homomorphism from $\langle S, \sigma\rangle$ to $\langle S', \sigma'\rangle$, and provides perhaps the most convincing argument that the notion of a bounded morphism is a natural one.

EXAMPLE 150. Let $\mathbb{X}$ and $\mathbb{X}'$ be two topological spaces, represented as coalgebras $\mathbb{X} = \langle X, \xi\rangle$ and $\mathbb{X}' = \langle X', \xi'\rangle$ for the filter functor $\mathcal{F}_{\breve{\mathcal{P}}}$ of Example 144. We leave it for the reader to check that a map $f : S \to S'$ is an $\mathcal{F}_{\breve{\mathcal{P}}}$-coalgebra homomorphism iff f is continuous and open (i.e., not only do we require $f^{-1}[U']$ to be open in $\mathbb{X}$ if U' is open in $\mathbb{X}'$, but also $f[U]$ must be open in $\mathbb{X}'$ for all $\mathbb{X}$-open U).

Likewise, one can prove that the coalgebraic notion of a homomorphism between monotone neighborhood frames, represented as coalgebras for the functor $\mathcal{U}_{\breve{\mathcal{P}}}$, corresponds to that of a bounded morphism for these structures as defined in section 2.

It is easy to check that the collection of coalgebra homomorphisms contains all identity arrows and is closed under arrow composition. Hence, the Ω-coalgebras with their homomorphisms form a category.

DEFINITION 151. For any functor $\Omega : \mathsf{C} \to \mathsf{C}$, we let $\mathsf{Coalg}(\Omega)$ denote the category with Ω-coalgebras as objects and the corresponding homomorphisms as arrows. The category C is called the *base category* of $\mathsf{Coalg}(\Omega)$.

The reader will already be familiar with a number of (isomorphic copies of) these categories. For instance, Example 149 shows in fact that the category Fr (of frames with bounded morphisms) is *isomorphic* to the category $\mathsf{Coalg}(\mathcal{P})$ of $\mathcal{P}$-coalgebras. Likewise, elaborating Example 147, one can prove that the category DGF (of descriptive general frames with continuous bounded morphisms, see Definition 66) is isomorphic to the category of Stone coalgebras for the Vietoris functor. Of course, it is these isomorphisms that justify our classification of modal structures as coalgebras, not so much the simple fact that the objects in isolation can be presented in coalgebraic format.

REMARK 152. Recall that an *algebra* over a signature Ω is a set A with an Ω-indexed collection $\{f^{\mathbb{A}} \mid A^{ar(f)} \to A\}$ of operations. These operations may be combined into a single map $\alpha : \sum_{f\in\Omega} A^{ar(f)} \to A$, where $\sum_{f\in\Omega} A^{ar(f)}$ denotes the *coproduct* (or *sum*, or *disjoint union*) of the sets $\{A^{ar(f)} \mid f \in \Omega\}$. It is not hard to verify that a map $g : A \to A'$ is an algebraic homomorphism between the algebras $\mathbb{A} = \langle A, \alpha\rangle$ and $\mathbb{A}' = \langle A', \alpha'\rangle$ iff the following diagram commutes:

$$\begin{tikzcd}
A \arrow[r, "f"] & A' \\
\Omega A \arrow[u, "\alpha"] \arrow[r, "\Omega f"] & \Omega A' \arrow[u, "\alpha'"']
\end{tikzcd}$$

where we now view the signature Ω as the polynomial set *functor* $\sum_{f\in\Omega} \mathcal{I}^{ar(f)}$. That is, Ω operates as well on functions between sets. This naturally suggests the following generalization.

Given an endofunctor Ω on a category C, an Ω-algebra is a pair $\mathbb{A} = \langle A, \alpha\rangle$ where $\alpha : \Omega A \to A$ is an arrow in C. A homomorphism from an Ω-algebra $\mathbb{A}$ to an Ω-algebra $\mathbb{A}'$ is an arrow $f : A \to A'$ such that $f \circ \alpha = \alpha' \circ (\Omega f)$. The induced category is denoted as $\mathsf{Alg}(\Omega)$.

Now the obvious similarities between the notions of algebra and coalgebra can be made very precise. The basic observation, which also explains the name 'coalgebra', is that a coalgebra $\mathbb{C} = \langle C, \gamma : C \to \Omega C\rangle$ over a base category C can also be seen as an algebra in the *opposite* category C^{op} — we will come back to this issue in section 15. Note however, that universal coalgebra, dealing with arbitrary set functors, is more general than (what is usually called) universal algebra, which involves only polynomial functors.

10 FINAL COALGEBRAS

DEFINITION 153. A functor $\Omega : \mathsf{C} \to \mathsf{C}$ is said to *admit a final* or *terminal coalgebra* if the category $\mathsf{Coalg}(\Omega)$ has a final object, that is, a coalgebra $\mathbb{Z}$ such that from every coalgebra $\mathbb{A}$ in $\mathsf{Coalg}(\Omega)$ there is a unique homomorphism $!_{\mathbb{A}} : \mathbb{A} \to \mathbb{Z}$.

Functors admitting a final coalgebra are of special interest. In the case of state-based coalgebras, one reason for this is that final coalgebras often provide an intuitive encoding of the notion of *behavior*. And in fact, many interesting and well-known mathematical objects can be naturally associated with the final coalgebra of some functor.

EXAMPLE 154. Consider a black box machine $\mathbb{M} = \langle M, \mu \rangle$ as in Example 141. Starting from, say, state x_0, the machine makes a transition $\mu(x_0) = (c_0, x_1)$ and continues with $\mu(x_1) = (c_1, x_2)$, $\mu(x_2) = (c_2, x_3)$, etc. Since the states x_0, x_1, ... are internal to the machine, the only observable part of this dynamics is the infinite sequence or *stream* $beh(x_0) = (c_0, c_1, c_2, \ldots) \in C^\omega$ of values in the data set C.

The collection C^ω of all infinite words over C forms itself a system for the functor $C \times \mathcal{I}$. Simply endow the set C^ω with the transition structure γ splitting an infinite stream $u = c_0c_1c_2\ldots$ into its *head* $h(u) = c_0$ and its *tail* $t(u) = c_1c_2c_3\ldots$ Putting $\gamma(u) = (h(u), t(u))$, one easily proves that the behavior map $x \mapsto beh(x)$ is the unique homomorphism from $\mathbb{M}$ to this coalgebra $\langle C^\omega, \gamma \rangle$. This shows that $\langle C^\omega, \gamma \rangle$ is the final object in the category $\mathsf{Coalg}(C \times \mathcal{I})$.

EXAMPLE 155. For a second example, consider again the coalgebraic representation of a deterministic automaton over the alphabet C as a $2 \times \mathcal{I}^C$-coalgebra. Now we will see that the collection $\mathcal{P}(C^*)$ of all *languages* over C provides (the carrier of) the final coalgebra. We can turn this set $\mathcal{P}(C^*)$ into a coalgebra by imposing on it the following transition function $\lambda : \mathcal{P}(C^*) \to 2 \times \mathcal{P}(C^*)^C$. Writing $\lambda(L) = (\lambda_0(L), \lambda_1(L))$, we define $\lambda_0(L) := 1$ iff the empty string belongs to L, and $\lambda_1(L)(c) := \{w \in C^* \mid cw \in L\}$. (The latter set is sometimes called the *c-derivative* of L.)

We leave it for the reader to verify that with this definition, the structure $\langle \mathcal{P}(C^*), \lambda \rangle$ forms the final object in $\mathsf{Coalg}(2 \times \mathcal{I}^C)$. Given a $2 \times \mathcal{I}^C$-coalgebra $\mathbb{A}$, the unique homomorphism $!_{\mathbb{A}} : \mathbb{A} \to \langle \mathcal{P}(C^*), \lambda \rangle$ maps a state $a \in A$ to the language that is accepted by the automaton that we obtain by taking a as initial state of $\mathbb{A}$.

EXAMPLE 156. An interesting example in modal logic is provided by the final coalgebra for the Vietoris functor $\mathcal{V}$ of Example 147. The existence of a final $\mathcal{V}$-coalgebra is in fact an immediate consequence of the isomorphism $\mathsf{Coalg}(\mathcal{V}) \cong \mathsf{DGF}$, and the duality between DGF and MA (the category of modal algebras with homomorphisms). MA has an initial object (namely, the Lindenbaum-Tarski algebra generated by the empty set of variables, or, equivalently, the free modal algebra over zero generators), and so by duality, $\mathsf{Coalg}(\mathcal{V})$ must have a final object. In fact, the *canonical* descriptive general frame, based on the set of maximal consistent closed formulas, fulfills this role — a nice and perhaps quite unexpected application of this construction.

An important application of final coalgebras is provided by the principle of *coinduction*, which is one of the fundamental coalgebraic notions. There are two sides to this principle: it serves both as an important proof tool and as an elegant means of providing definitions. As a definition principle, coinduction is based on the existence of unique homomorphisms into the final Ω-system $\mathbb{Z} = \langle Z, \zeta \rangle$. For, suppose that we can endow a set S with an Ω-

coalgebra map $\sigma : S \to \Omega S$, thus obtaining the Ω-system $\mathbb{S}$. Then there is a unique function $f_\sigma = !_{\mathbb{S}} : S \to Z$ which is consistent with the coalgebra specification σ, in the sense that it is a coalgebraic homomorphism from $\langle S, \sigma \rangle$ to $\mathbb{Z}$. Thus the function f_σ is *defined* by coinduction from (the specification) σ.

EXAMPLE 157. For instance, take the function that merges two streams by taking elements from either stream in turn. For a coinductive definition of this map, define the transition map $zip : C^\omega \times C^\omega \to C \times (C^\omega \times C^\omega)$ as follows:

$$zip(u, v) := (h(u), (v, t(u))),$$

where h and t are the head and tail maps of Example 154. Then by finality there is a unique homomorphism $f_{zip} : C^\omega \times C^\omega \to C^\omega$. One may verify that this indeed defines the map that zips two streams together.

The previous example is fairly typical in that it uses coinduction to define a function from a product of the final system to itself. It should also be noted that coinduction works particularly well for structures that combine algebraic and coalgebraic features, such as streams of data objects which are subject themselves to algebraic operations.

Unfortunately, final coalgebras do not exist for every functor Ω. For instance, Set-endofunctors involving the power set functor in a nontrivial way, will generally not admit a final coalgebra; in particular, there is no final Kripke frame or model. By Cantor's theorem, these results are immediate consequence of the following proposition, which is due to Lambek [79].

PROPOSITION 158. *Let $\Omega : \mathsf{C} \to \mathsf{C}$ be some functor admitting a final system $\mathbb{Z} = \langle Z, \zeta \rangle$. Then ζ is an isomorphism (in C) between Z and ΩZ.*

Proof. Suppose that $\mathbb{Z} = \langle Z, \zeta \rangle$ is the final object of $\mathsf{Coalg}(\Omega)$. It can easily be verified that ζ is in fact a coalgebra homomorphism from $\mathbb{Z}$ to $\mathbb{Z}_2 := \langle \Omega Z, \Omega\zeta \rangle$. But then the composition $!_{\mathbb{Z}_2} \circ \zeta$ is a coalgebra homomorphism from $\mathbb{Z}$ to itself, just like the identity arrow id_Z on Z. Thus by uniqueness it follows that $!_{\mathbb{Z}_2} \circ \zeta = id_Z$. For the reverse composition $\zeta \circ !_{\mathbb{Z}_2}$ we have, by the fact that $!_{\mathbb{Z}_2}$ is a homomorphism, that $\zeta \circ !_{\mathbb{Z}_2} = \Omega !_{\mathbb{Z}_2} \circ \Omega\zeta = \Omega(!_{\mathbb{Z}_2} \circ \zeta) = \Omega(id_Z) = id_{\Omega Z}$. From this the result is immediate. ❑

So which functors admit final coalgebras? Some good sufficient conditions are known.

DEFINITION 159. Let Ω be some set functor, and κ some cardinal. Call Ω *κ-small* if

$$\Omega(S) = \bigcup \{(\Omega\iota)[\Omega(A)] \mid \iota : A \hookrightarrow S, |A| < \kappa\},$$

for all sets $S \neq \varnothing$. Ω is *small* if it is small for some cardinal κ.

In words, the definition requires every element of $\Omega(S)$ to be in the range of $\Omega\iota$ for an appropriate inclusion map $\iota : A \to S$. In case Ω is a *standard* functor (meaning that Ω maps inclusions $\iota : A \hookrightarrow B$ to inclusions $(\Omega\iota) : \Omega A \hookrightarrow \Omega B$), the definition boils down to the requirement that $\Omega(S) = \bigcup\{\Omega(A) \mid A \subseteq S, |A| < \kappa\}$. The notion of smallness is easily seen to be equivalent to the instantiation in Set of the more general notion of *accessibility*, and it is also equivalent to the concept of *boundedness*, cf. Adámek & Porst [6] for details.

Examples of small functors abound; for instance, whenever we replace, in a Kripke polynomial functor, the power set functor by a bounded variant such as the finite power

set functor, the result is a small functor. For instance, the finite power set functor $\mathcal{P}_\omega$ is ω-small. The following result, due to Aczel & Mendler [3] and Barr [9], witnesses the importance of the notion.

FACT 160. Every small set functor admits a final coalgebra.

As one of the immediate corollaries of this fact, the categories of image finite frames and image finite models, which can be represented as coalgebras for the functor $\mathcal{P}_\omega$, and $\mathcal{P}\mathsf{Prop} \times \mathcal{P}_\omega$, respectively, have final objects.

REMARK 161. For Set-based functors that do not admit a final coalgebra, one may *create* a final coalgebra — at least, if one is willing to allow coalgebras with a class rather than a set as their carrier. Let SET be the category that has classes as objects, and set-continuous functions as arrows. These are functions $f : \mathsf{C} \to \mathsf{C}'$ between classes with the property that $f(\mathsf{C}) = \bigcup\{f(S) \mid S \subseteq \mathsf{C}$ and S is a set$\}$. An endofunctor on SET is *set-based* if for each class C and each $c \in \Omega(C)$ there is a set $S \subseteq C$ such that $c \in (\Omega\iota)[\Omega(S)]$, where $\iota : S \to C$ is the inclusion map. (If the set functor is standard, this boils down to requiring that Ω is a set-continuous map on objects.) Now Aczel & Mendler [3] proved that every set-based endofunctor $\Xi : \mathsf{SET} \to \mathsf{SET}$ admits a final coalgebra. The similarity to Fact 160 is no coincidence: Barr [9] showed that the result of Aczel & Mendler can in fact be reformulated as Fact 160.

This fact can be used as follows. Given an endofunctor Ω on Set, there is a *unique* way to extend Ω to a set based endofunctor Ω^+ on SET. (On objects, simply put $\Omega^+(\mathsf{C}) := \bigcup\{(\Omega\iota)[\Omega(S) \mid \iota : S \hookrightarrow \mathsf{C}, S \text{ a set}\}$.) The theorem of Aczel & Mendler then guarantees the existence of a final object $\mathbb{Z}$ in $\mathsf{Coalg}(\Omega^+)$. This coalgebra will be class-based if Ω does not admit a final coalgebra, but it will be final, not only with respect to the set-based coalgebras in $\mathsf{Coalg}(\Omega^+)$, but also with respect to the class-based ones. As an important instance of this idea, Aczel [2] showed that the class of non-well-founded sets provides the final coalgebra for (the SET-based extension of) the power set functor.

REMARK 162. Whether the functor admits a final coalgebra or not, one may always (try to) approximate it. The *final* or *terminal sequence* associated with a given set functor Ω, is an ordinal indexed sequence of objects $\langle Z_\alpha \rangle$ with maps $p^\alpha_\beta : Z_\alpha \to Z_\beta$ for $\beta \leq \alpha$, such that (i) $Z_{\alpha+1} = \Omega Z_\alpha$ and $p^{\alpha+1}_{\beta+1} = \Omega p^\alpha_\beta$, (ii) $p^\alpha_\alpha = id_{Z_\alpha}$ and $p^\beta_\gamma \circ p^\alpha_\beta = p^\alpha_\gamma$, (iii) if λ is a limit ordinal, then Z_λ with $\{p^\lambda_\alpha \mid \alpha < \lambda\}$ is a limit of the diagram with objects $\{Z_\alpha \mid \alpha < \lambda\}$ and arrows $\{p^\alpha_\beta \mid \alpha, \beta < \lambda\}$. (In particular, taking 0 to be a limit ordinal, we find that $Z_0 = 1$ is some initial object 1 of the category Set.) It is not hard to prove that, modulo isomorphism, the final sequence is uniquely determined by these conditions.

Intuitively, it can be seen as an approximation of the final coalgebra for Ω. That is, where elements of the final coalgebra represent 'complete' behavior, elements of Z_α represent behavior that can be performed in α steps. To make this precise and formal, observe that for any Ω-coalgebra $\mathbb{S}$ there is a *unique* ordinal-indexed class of functions $!_\alpha : S \to Z_\alpha$ such that $!_0$ is fixed by the finality of Z_0 in Set, $!_{\alpha+1} = (\Omega !_\alpha) \circ \sigma$, and for limit λ, $!_\lambda$ is given as the unique map $!_\lambda : S \to Z_\lambda$ such that $!_\lambda = p^\lambda_\alpha \circ !_\alpha$ for all $\alpha < \lambda$. It is not hard to prove that, for instance, $\mathbb{S}, s \equiv_\Omega \mathbb{S}', s'$ implies that $!_\alpha(s) = !_\alpha(s')$ for all α.

The relation with final coalgebras can be made precise, as follows. On the one hand, if the final sequence converges, in the sense that some arrow $p^{\alpha+1}_\alpha$ is a bijection, then the coalgebra $\langle Z_\alpha, (p^{\alpha+1}_\alpha)^{-1}\rangle$ is a final coalgebra for Ω. And conversely, under some constraints on Ω, Adámek & Koubek [5] proved that if Ω admits a final coalgebra, then

the final sequence converges to it. More information on the final sequence of set functors can be found in Worrell [109].

11 BISIMULATION & BEHAVIORAL EQUIVALENCE

In this section we discuss the most important notions of equivalence between systems: behavioral equivalence and bisimulation. Both of these generalize the concept of a bisimulation between two Kripke models.

Probably the most intuitive notion of equivalence between systems is that of *behavioral*, or *observational*, equivalence. The idea here is to consider two states to be similar if we cannot distinguish them by observations, because they display the same behavior. For instance, we call two deterministic automata (pointed $2 \times \mathcal{I}^C$-coalgebras) equivalent if they recognize the same language. In case the functor Ω admits a final coalgebra $\mathbb{Z}$, this idea is easily formalized by making state s_0 in coalgebra $\mathbb{S}_0$ equivalent to state s_1 in coalgebra $\mathbb{S}_1$ if $!_{\mathbb{S}_0}(s_0) = !_{\mathbb{S}_1}(s_1)$. In case the functor does not admit a final coalgebra, we generalize this demand as follows.

DEFINITION 163. Let $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$ be two systems for the set functor Ω. Then $s \in S$ and $s' \in S'$ are *behaviorally equivalent*, notation: $\mathbb{S}, s \equiv_\Omega \mathbb{S}', s'$ if there is an Ω-system $\mathbb{X} = \langle X, \xi \rangle$ and homomorphisms $f : \mathbb{S} \to \mathbb{X}$ and $f' : \mathbb{S}' \to \mathbb{X}$ such that $f(s) = f'(s')$.

REMARK 164. It is easily checked that in case Ω admits a final coalgebra, then indeed $\mathbb{S}, s \equiv_\Omega \mathbb{S}', s'$ iff $!_{\mathbb{S}}(s) = !_{\mathbb{S}'}(s')$. In the case that Ω does not admit a final coalgebra, then one may show that behavioral equivalence is captured in the same way by the final coalgebra of the extension Ω^+ of Ω to the category SET, see Remark 161.

REMARK 165. As a variation of behavioral equivalence, the final sequence can be used to study behavior, in a way that is not unlike modal logic. For instance, call two pointed Ω-systems $(\mathbb{S}, s)$ and $(\mathbb{S}', s')$ *α-equivalent* if $!_\alpha(s) = !_\alpha(s')$. In the case of Kripke models, this notion coincides with that of bounded bisimilarity, see Chapter 5 of this volume. One may prove that behavioral equivalence itself coincide with the intersection of α-equivalence for all ordinals α.

In almost all cases of interest, behavioral equivalence can be characterized via the equally fundamental concept of *bisimilarity*, which is due to Aczel & Mendler [3]. The definition of bisimilarity and bisimulations may not be so intuitive at first sight, but, as we will see, these notions have some rather elegant mathematical properties.

DEFINITION 166. Let $\mathbb{S} = \langle S, \sigma \rangle$ and $\mathbb{S}' = \langle S', \sigma' \rangle$ be two systems for the set functor Ω. A relation $B \subseteq S \times S'$ is called a *bisimulation between* $\mathbb{S}$ and $\mathbb{S}'$, if we can endow it with a coalgebra map $\beta : B \to \Omega B$, in such a way that the two projections $\pi : B \to S$ and $\pi' : B \to S'$ are homomorphisms from $\langle B, \beta \rangle$ to $\mathbb{S}$ and $\mathbb{S}'$, respectively:

$$
\begin{tikzcd}
S \arrow[d, "\sigma"'] & B \arrow[l, "\pi"'] \arrow[r, "\pi'"] \arrow[d, dotted, "\beta"'] & S' \arrow[d, "\sigma'"'] \\
\Omega S & \Omega B \arrow[l, "\Omega\pi"'] \arrow[r, "\Omega\pi'"] & \Omega S'
\end{tikzcd}
$$

If there exists a bisimulation B with $(s, s') \in B$, we say that s and s' are *bisimilar*,

notation: $\mathbb{S}, s \leftrightarroweq \mathbb{S}', s'$ (or $B : \mathbb{S}, s \leftrightarroweq \mathbb{S}', s'$ in case we want to make the bisimulation B explicit).

Finally, if $\mathbb{S} = \mathbb{S}'$ we say that B is a bisimulation *on* $\mathbb{S}$; if this B happens to be an equivalence relation, we call it a *bisimulation equivalence* on $\mathbb{S}$.

REMARK 167. Intuitively, bisimulation equivalences correspond to *congruences* in universal algebra. To make this analogy somewhat more precise, call a relation $R \subseteq A_0 \times A_1$, linking the carrier sets of two Ω-algebras $\mathbb{A}_0$ and $\mathbb{A}_1$, *substitutive* if there exists an *algebraic structure* $\rho : \Omega R \to R$, such that the two projections $\pi_i : R \to A_i$ are (algebraic) homomorphisms. This is clearly an algebraic *analogue* (rather than a dual version) of a bisimulation, so that the correspondence between congruences and bisimulation equivalences obtains through the observation that a congruence is nothing but a substitutive equivalence relation.

EXAMPLE 168. Let $\mathbb{S}_0 = \langle S_0, \sigma_0 \rangle$ and $\mathbb{S}_1 = \langle S_1, \sigma_1 \rangle$ be two coalgebras over the functor $\mathcal{P}\mathsf{Prop} \times \mathcal{P}$. That is, $\mathbb{S}_0$ and $\mathbb{S}_1$ are Kripke models in coalgebraic shape; write $\sigma_i(s) = (\lambda_i(s), R_i[s])$, where $\lambda_i(s)$ is the collection of proposition letters true at s in $\mathbb{S}_i$, and $R_i[s]$ is the successor set of s in $\mathbb{S}_i$, as in the examples 143 and 149. Now consider an arbitrary relation $B \subseteq S_0 \times S_1$. It is a very instructive exercise to check that B is a bisimulation in the coalgebraic sense if and only if it is a bisimulation in the sense of Kripke models. Recall that the latter property means that for any pair $(s_0, s_1) \in B$:

(atom) $p \in \lambda_0(s)$ iff $p \in \lambda_1(s)$, for all $p \in \mathsf{Prop}$;

(forth) for all $t_0 \in R_0[s_0]$ there is some $t_1 \in R_1[s_1]$ with $(t_0, t_1) \in B$;

(back) for all $t_1 \in R_1[s_1]$ there is some $t_0 \in R_0[s_0]$ with $(t_1, t_0) \in B$.

One way to prove this equivalence uses the fact that bounded morphisms coincide with coalgebra morphisms, cf. Example 149. Details are left to the reader.

EXAMPLE 169. Recall from Example 142 that deterministic automata over an alphabet C can be represented as $2 \times \mathcal{I}^C$-coalgebras. Now let $\mathbb{A} = (A, o, \nu)$ and $\mathbb{A}' = (A', o', \nu')$ be two such automata. We leave it for the reader to verify that $B \subseteq A \times A'$ is a bisimulation between $\mathbb{A}$ and $\mathbb{A}'$ iff every pair $(s, s') \in B$ satisfies (i) $o(s) = o'(s')$ and (ii) $(\nu(s)(c), \nu'(s')(c)) \in B$ for every $c \in C$. In this case it is easy to see that bisimilar states are also behaviorally equivalent.

EXAMPLE 170. For an arbitrary set functor Ω, it is easy to see that for any coalgebra $\mathbb{S}$, the diagonal relation Δ_S is a bisimulation equivalence on $\mathbb{S}$. Furthermore, the converse of a bisimulation is again a bisimulation. However, the collection of bisimulations is not in general closed under taking relational composition.

Finally, homomorphisms can be seen as functional bisimulations. To be more precise, let $f : S_0 \to S_1$ be a function between the carriers of two Ω-coalgebras $\mathbb{S}_0$ and $\mathbb{S}_1$. Recall that the *graph* of f is the relation $G_f := \{(s, f(s)) \mid s \in S\}$. Then it holds that

$$f \text{ is a coalgebraic homomorphism iff its graph is a bisimulation.} \tag{42}$$

In order to see why this is so, first suppose that $G_f : \mathbb{S}_0 \leftrightarroweq \mathbb{S}_1$. Since the projection map $\pi_0 : G_f \to S_0$ is a bijective homomorphism, its inverse π_0^{-1} is also a homomorphism. But then $f = \pi_1 \circ \pi_0^{-1}$, as the composition of two homomorphisms, is also a homomorphism. For the other direction, suppose that f is a homomorphism; then it is straightforward

to verify that the map $(\Omega\pi_0)^{-1} \circ \sigma \circ \pi_0$ equips the set G_f with the required coalgebraic structure.

Bisimulations admit an elegant alternative characterization which involves the notion of *relation lifting*. As an example, consider the power set functor $\mathcal{P}$. Recall that $B \subseteq S_0 \times S_1$ is a bisimulation between $\mathbb{S}_0 = \langle S_0, R_0[\cdot]\rangle$ and $\mathbb{S}_1 = \langle S_1, R_1[\cdot]\rangle$ iff B satisfies the conditions *(back)* and *(forth)* of Example 168. Now suppose that we define, for an arbitrary relation $R \subseteq S_0 \times S_1$, the relation $\overline{\mathcal{P}}(R) \subseteq \mathcal{P}(S_0) \times \mathcal{P}(S_1)$ by putting

$$\overline{\mathcal{P}}(R) := \{(Q_0, Q_1) \mid \forall q_0 \in Q_0\, \exists q_1 \in Q_1.\, (q_0, q_1) \in R \text{ and } \forall q_1 \in Q_1\, \exists q_0 \in Q_0.\, (q_0, q_1) \in R\}. \tag{43}$$

In other words, we *lift* the relation R to the level of the power sets of S_0 and S_1. The definition of a bisimulation between $\mathcal{P}$-coalgebras can now be nicely characterized as follows:

$$B : \mathbb{S}_0 \leftrightarroweq \mathbb{S}_1 \text{ iff } (R_0[s_0], R_1[s_1]) \in \overline{\mathcal{P}}(B) \text{ for all } (s_0, s_1) \in B.$$

This nice way of characterizing bisimulation via relation lifting is not limited to the power set functor — it applies in fact to *every* set functor.

DEFINITION 171. Let $\mathbb{S}_0$ and $\mathbb{S}_1$ be two coalgebras for some set functor Ω. Given a relation $R \subseteq S_0 \times S_1$, consider the following diagram, where $\pi_i : R \to S_i$ and $p_i : \Omega S_0 \times \Omega S_1 \to \Omega S_i$ denote the projection maps.

$$\begin{tikzcd}
S_0 \arrow[d, "\sigma_0"'] & R \arrow[l, "\pi_0"'] \arrow[r, "\pi_1"] & S_1 \arrow[d, "\sigma_1"] \\
\Omega S_0 & \Omega R \arrow[l, "\Omega\pi_0"'] \arrow[r, "\Omega\pi_1"] \arrow[d, dotted, "\rho_R"'] & \Omega S_1 \\
 & \Omega S_0 \times \Omega S_1 \arrow[ul, "p_0"] \arrow[ur, "p_1"'] &
\end{tikzcd}$$

It follows from the category theoretic properties of the product $\Omega S_0 \times \Omega S_1$ that there is a unique map $\rho_R = \langle \Omega\pi_0, \Omega\pi_1\rangle$ from ΩR to $\Omega S_0 \times \Omega S_1$ such that $p_i \circ \rho_R = \Omega\pi_i$ for $i = 0, 1$. We define the *relation lifting* of R as the relation

$$\overline{\Omega} R := \{((\Omega\pi_0)(u), (\Omega\pi_1)(u)) \mid u \in \Omega R\}, \tag{44}$$

that is, $\overline{\Omega} R$ is the image of ΩR under ρ_R.

The results listed in the following theorem, which summarize the most important properties of bisimulations, basically date back to Aczel & Mendler [3].

THEOREM 172. *Let $\mathbb{S}_0$ and $\mathbb{S}_1$ be two coalgebras for some set functor Ω.*

1. *$B : \mathbb{S}_0 \leftrightarroweq \mathbb{S}_1$ iff $(\sigma_0(s_0), \sigma_1(s_1)) \in \overline{\Omega}(B)$ for all $(s_0, s_1) \in B$.*

2. *The collection of bisimulations between $\mathbb{S}_0$ and $\mathbb{S}_1$ forms a complete lattice under the inclusion order, with joins given by unions.*

3. *The bisimilarity relation $\leftrightarroweq$ is the largest bisimulation between $\mathbb{S}_0$ and $\mathbb{S}_1$.*

Proof. The first part of the theorem is an almost immediate consequence of the definitions, so we leave the details to the reader.

The crucial observation in the proof of the other two parts is that

$$\overline{\Omega} : \mathcal{P}(S_0 \times S_1) \to \mathcal{P}(\Omega S_0 \times \Omega S_1) \text{ is a monotone operation.} \tag{45}$$

For a proof, let $R \subseteq R'$ be two relations between S_0 and S_1, with $\iota : R \to R'$ denoting the inclusion map. By definition of $\overline{\Omega}$, we may without loss of generality represent an arbitrary element of $\overline{\Omega}(R)$ as a pair $\rho_R(u) = ((\Omega\pi_0)(u), (\Omega\pi_1)(u))$ for some $u \in \Omega R$. Define $u' := (\Omega\iota)(u)$, then u' belongs to $\Omega R'$, and for each i we find that $(\Omega\pi'_i)(u') = (\Omega\pi'_i \circ \Omega\iota)(u) = (\Omega(\pi'_i \circ \iota)(u) = (\Omega\pi_i)(u)$. That is, $\rho_R(u) = \rho_{R'}(u')$, which shows that $\rho_R(u)$ belongs to $\overline{\Omega}R'$. This proves (45).

Now for the proof of part 2, recall that a partial order is a complete lattice if it closed under arbitrary joins. Hence, it suffices to prove that the union B of a collection $\{B_j \mid j \in J\}$ of bisimulations is again a bisimulation. Take an arbitrary pair $(s_0, s_1) \in B$. Then (s_0, s_1) belongs to B_j for some $j \in J$. Hence, by part 1, we find (s_0, s_1) in $\overline{\Omega}(B_j)$, so $(s_0, s_1) \in \overline{\Omega}(B)$ by the monotonicity of $\overline{\Omega}$. But then B is a bisimulation by part 1.

Finally, for part 3, note that it is an immediate consequence of part 2 that $\leftrightarroweq$, being the union of all bisimulations between $\mathbb{S}_0$ and $\mathbb{S}_1$, is a bisimulation itself. Hence, by definition, it is the greatest bisimulation between $\mathbb{S}_0$ and $\mathbb{S}_1$. In fact, it follows by the Knaster-Tarski theorem (on fixed points of monotone operations on complete lattices), that $\leftrightarroweq$ is in fact the greatest fixed point of the map $\Lambda : R \mapsto \{(s_0, s_1) \mid (\sigma_0(s_0), \sigma_1(s_1)) \in \overline{\Omega}(R)\}$. ❑

In the case of Kripke polynomial functors, relation lifting can be characterized using *induction* on the construction of the functor, cf. Jacobs [63].

PROPOSITION 173. *Let S and S' be two sets, and $R \subseteq S \times S'$ a binary relation between S and S'. Then the following induction defines the relation lifting $\overline{K}(R) \subseteq KS \times KS'$, for each Kripke polynomial functor K:*

$$\begin{array}{rcl}
\overline{\mathcal{I}}(R) & := & R, \\
\overline{C}(R) & := & \Delta_C, \\
\overline{K_0 \times K_1}(R) & := & \{((x_0, x_1), (x'_0, x'_1)) \mid (x_0, x'_0) \in \overline{K_0}(R) \text{ and } (x_1, x'_1) \in \overline{K_1}(R)\}, \\
\overline{K_0 + K_1}(R) & := & \{(\kappa_0 x_0, \kappa_0 x'_0) \mid (x_0, x'_0) \in \overline{K_0}(R)\} \cup \{(\kappa_1 x_1, \kappa_1 x'_1) \mid (x_1, x'_1) \in \overline{K_1}(R)\}, \\
\overline{K^D}(R) & := & \{(f, f') \mid (f(d), f'(d)) \in \overline{K}(R) \text{ for all } d \in D\}, \\
\overline{\mathcal{P}K}(R) & := & \{(Q, Q') \mid \forall q \in Q\, \exists q' \in Q'.\, (q, q') \in \overline{K}(R) \text{ and} \\
& & \qquad \forall q' \in Q'\, \exists q \in Q.\, (q, q') \in \overline{K}(R)\}.
\end{array}$$

Now that we have defined these two notions of equivalence between coalgebras, the obvious question is how they relate to each other. One direction is clear: bisimilarity is a sufficient condition for behavioral equivalence.

PROPOSITION 174. *Let $\Omega : \mathsf{Set} \to \mathsf{Set}$ be some functor, and let s_0 and s_1 be states of the Ω-coalgebras $\mathbb{S}_0$ and $\mathbb{S}_1$, respectively. Then $\mathbb{S}_0, s_0 \leftrightarroweq \mathbb{S}_1, s_1$ implies $\mathbb{S}_0, s_0 \equiv_\Omega \mathbb{S}_1, s_1$.*

Proof. The proof of this proposition is, in the general case, similar to the one of Theorem 177 below (with an application of *pushouts* instead of pullbacks), so we omit details.

In the special case that Ω admits a final coalgebra, a very simple proof obtains. Assume that $B : \mathbb{S}_0 \leftrightarroweq \mathbb{S}_1$, and let $\beta : B \to \Omega B$ be a coalgebra map witnessing this. It follows from the definitions that both $!_{\mathbb{S}_0} \circ \pi_0$ and $!_{\mathbb{S}_1} \circ \pi_1$ are coalgebraic homomorphisms from $\langle B, \beta \rangle$ to the final coalgebra, so from finality it follows that $!_{\mathbb{S}_0} \circ \pi_0 = !_{\mathbb{S}_1} \circ \pi_1$. From this it is immediate that $B \subseteq \equiv_\Omega$. Hence in particular, since $\leftrightarroweq$ is itself a bisimulation, we see that $\leftrightarroweq \subseteq \equiv_\Omega$. ❑

In general however, bisimilarity is a strictly stronger notion than behavioral equivalence. For instance, the two notions do not coincide in the case of monotone neighborhood frames (coalgebras for the functor $\mathcal{U}_{\breve{\mathcal{P}}}$ of Example 144). The reader is referred to Hansen & Kupke [55] for details. Here we just mention that behavioral equivalence, which for monotone neighborhood frames is formulated exactly like the *topobisimilarity* defined in Chapter 1 of this volume, seems to be the more natural notion.

For a constraint on the functor that guarantees the two notions to coincide, consider the following.

DEFINITION 175. A *weak pullback* of two arrows $f_0 : A_0 \to B$, $f_1 : A_1 \to B$ in a category C is a pair of arrows $p_0 : W \to A_0$, $p_1 : W \to A_1$ such that (i) $f_0 \circ p_0 = f_1 \circ p_1$, while (ii) for every pair $p_0' : W' \to A_0$, $p_1' : W' \to A_1$ that also satisfies $f_0 \circ p_0' = f_1 \circ p_1'$, there is a mediating arrow $w' : W' \to W$ such that $p_0 \circ w' = p_0'$ and $p_1 \circ w' = p_1'$.
A functor $\Omega : \mathsf{C} \to \mathsf{C}'$ *preserves weak pullbacks* if for any weak pullback (p_0, p_1) of any (f_0, f_1) in C, the pair $(\Omega p_0, \Omega p_1)$ is a weak pullback of $(\Omega f_0, \Omega f_1)$ in C'.

$$\begin{tikzcd}
& A_0 \arrow[r, "f_0"] & B \\
& W \arrow[u, "p_0"] \arrow[r, "p_1"] & A_1 \arrow[u, "f_1"'] \\
W' \arrow[uur, "p_0'"] \arrow[urr, "p_1'"'] \arrow[ur, dotted]
\end{tikzcd}$$

Note that the mediating arrow w' need not be unique: adding this requirement to the definition would give the more familiar, and stronger, notion of a *pullback*. The category Set has pullbacks: for $f_0 : A_0 \to B$ and $f_1 : A_1 \to B$, we can take the projections to A_0 and A_1 from the set $pb(f_0, f_1) := \{(a_0, a_1) \in A_0 \times A_1 \mid f_0(a_0) = f_1(a_1)\}$.

Many but not all endofunctors on Set in fact preserve weak pullbacks.

PROPOSITION 176. *All polynomial functors preserve pullbacks, and all Kripke polynomial functors preserve weak pullbacks.*

This prima facie rather exotic property is of great importance in the theory of universal coalgebra. The main reason for this is that Ω preserving weak pullbacks is equivalent to $\overline{\Omega}$ commuting with relational composition, that is, satisfying $\overline{\Omega}(R \circ R') = \overline{\Omega}(R) \circ \overline{\Omega}(R')$. In fact, one may show that any set functor Ω preserves weak pullbacks if and only if $\overline{\Omega}$ is an endofunctor on the category with sets as objects and binary relations as arrows. This result is often attributed to Carboni, Kelly & Wood [19], but it already follows from earlier work by Trnková [104, 105] and Barr [10]. In any case, the importance of the notion in the theory of coalgebras lies in the results from Rutten [97] that are given in the next theorem.

THEOREM 177. *Assume that the functor* $\Omega : \mathsf{Set} \to \mathsf{Set}$ *preserves weak pullbacks. Then the collection of bisimulations is closed under taking relational composition, and the notions of bisimilarity and behavioral equivalence coincide.*

Proof. We leave the proof of the first statement as an exercise for the reader, and concentrate on the second statement. Let s_0 and s_1 be states of the Ω-coalgebras $\mathbb{S}_0$ and $\mathbb{S}_1$, respectively. We need to prove that $\mathbb{S}_0, s_0 \leftrightarroweq \mathbb{S}_1, s_1$ iff $\mathbb{S}_0, s_0 \equiv_\Omega \mathbb{S}_1, s_1$. Because of Proposition 174 it suffices to prove the direction from right to left.

Let $f_0 : \mathbb{S}_0 \to \mathbb{X}$ and $f_1 : \mathbb{S}_1 \to \mathbb{X}$ be two homomorphisms such that $f_0(s_0) = f_1(s_1)$. Then in Set, the set $B := \{(s_0, s_1) \in S_0 \times S_1 \mid f_0(s_0) = f_1(s_1)\}$, together with the projection functions $\pi_0 : B \to S_0$ and $\pi_1 : B \to S_1$ constitutes a pullback of f_0 and f_1, cf. the square in the foreground of the picture. Because Ω preserves weak pullbacks, the diagram in the background of the picture is a weak pullback diagram in Set.

Now consider the two arrows $\sigma_i \circ \pi_i : B \to \Omega(S_i)$. First observe that $\Omega f_i \circ \sigma_i = \xi \circ f_i$ for each i, because each f_i is a coalgebra homomorphism. Hence, chasing the diagram we find that

$$\Omega f_0 \circ \sigma_0 \circ \pi_0 = \xi \circ f_0 \circ \pi_0 = \xi \circ f_1 \circ \pi_1 = \Omega f_1 \circ \sigma_1 \circ \pi_1.$$

Since $\Omega\pi_0$ and $\Omega\pi_1$ form a weak pullback of Ωf_0 and Ωf_1, this implies the existence of a mediating function $\beta : B \to \Omega B$ such that $\Omega\pi_i \circ \beta = \sigma_i \circ \pi_1$. In other words, $\mathbb{B} := \langle B, \beta \rangle$ is an Ω-coalgebra, and the projection maps π_0 and π_1 are homomorphisms from $\mathbb{B}$ to $\mathbb{S}_0$ and $\mathbb{S}_1$, respectively.

$$\begin{tikzcd}
 & \Omega S_0 \arrow[rr, "\Omega f_0"] & & \Omega X \\
S_0 \arrow[ur, "\sigma_0"] \arrow[rr, "f_0"] & & X \arrow[ur, "\xi"'] & \\
 & \Omega B \arrow[uu, "\Omega\pi_0"] \arrow[rr, "\Omega\pi_1"'] & & \Omega S_1 \arrow[uu, "\Omega f_1"'] \\
B \arrow[uu, "\pi_0"] \arrow[ur, dotted, "\beta"'] \arrow[rr, "\pi_1"'] & & S_1 \arrow[uu, "f_1"'] \arrow[ur, "\sigma_1"'] &
\end{tikzcd}$$

❑

We finish the section with a brief discussion of *coinduction* as a coalgebraic proof principle. This principle states, for a system $\mathbb{S}$, that $\leftrightarroweq \subseteq \Delta_S$; or equivalently, that every bisimulation is a subset of the diagonal Δ_S. The importance of this principle is that, when applicable to $\mathbb{S}$, in order to prove the identity of two states in $\mathbb{S}$, it suffices to show that they are linked by some bisimulation. It is not hard to prove that *final coalgebras*, if existing, satisfy the principle of coinduction. This principle has surprisingly powerful applications. For instance, since the class of non-well-founded sets is (in $\mathsf{Coalg}(\mathcal{P}^+)$, cf. Remark 161) the final coalgebra of the power set functor, bisimilarity may serve as a notion of identity between sets, see Aczel [2]. As a second example, Rutten [96] is a presentation of the theory of deterministic automata and (regular) languages in which coinduction on the final coalgebra of Example 155 is the basic proof principle.

12 COVARIETIES

What is the coalgebraic analog of a variety? In other words, what are natural closure operations on classes of coalgebras? We start with homomorphic images.

DEFINITION 178. Let Ω be some endofunctor on Set. If $\varphi : \mathbb{A} \to \mathbb{B}$ is a surjective homomorphism between the Ω-coalgebras $\mathbb{A}$ and $\mathbb{B}$, then we say that $\mathbb{B}$ is a *homomorphic image* of $\mathbb{A}$.

In universal algebra, one finds a one-one correspondence between homomorphic images and congruences. Something similar applies here, but the analogy is perfect only in the case of functors that preserve weak pullbacks.

PROPOSITION 179. *Let $\mathbb{S} = \langle S, \sigma \rangle$ be an Ω-coalgebra for some set functor Ω. Then*

1. *Given a bisimulation equivalence E on $\mathbb{S}$, there is a unique coalgebra structure σ' on S/E such that the quotient map $\nu : S \to S/E$ is a homomorphism.*
2. *If Ω preserves weak pullbacks, then $\ker(\varphi)$ is a bisimulation equivalence for any homomorphism $\varphi : \mathbb{S} \to \mathbb{S}'$.*

Proof. For part 1, the coalgebra map σ' can be defined by putting $\sigma'([s]_E) := (\Omega\nu)\circ\sigma(s)$. Further proof details can be found in Rutten [97]. For the second part of the proposition, observe that $\ker(\varphi)$ is the relational composition of the graph of φ with its converse. The result then follows from Theorem 177. ❑

The next class operation that we consider is that of taking subcoalgebras.

DEFINITION 180. Let $\mathbb{A} = \langle A, \alpha \rangle$ and $\mathbb{S} = \langle S, \sigma \rangle$ be two Ω-coalgebras, such that S is a subset of A. If the inclusion map $\iota : S \to A$ is a homomorphism from $\langle S, \sigma \rangle$ to $\langle A, \alpha \rangle$, then we say that S is *open* with respect to $\mathbb{A}$, and we call the structure $\langle S, \alpha{\restriction}_S \rangle$ a *subcoalgebra* of $\mathbb{A}$.

Interestingly enough, the transition map of a subcoalgebra is completely determined by the underlying open set:

PROPOSITION 181. *Let $\mathbb{S}_0 = \langle S, \sigma_0 \rangle$ and $\mathbb{S}_1 = \langle S, \sigma_1 \rangle$ be two subcoalgebras of the coalgebra $\mathbb{A}$. Then $\sigma_0 = \sigma_1$.*

Proof. The case of S being empty is trivial, so suppose otherwise. Then from the assumption that $\mathbb{S}_0$ and $\mathbb{S}_1$ are subcoalgebras of $\mathbb{A}$, we may infer that $(\Omega\iota) \circ \sigma_0 = \alpha \circ \iota = (\Omega\iota) \circ \sigma_1$, where ι is the inclusion map of S into A. It follows from the functoriality of Ω that $\Omega\iota$ is an injection, so that we may conclude that $\sigma_0 = \sigma_1$. ❑

Some further observations concerning subcoalgebras are in order. First of all, the topological terminology is justified by the following proposition.

PROPOSITION 182. *Given a coalgebra $\mathbb{A}$ for some set functor Ω, the collection $\tau_{\mathbb{A}}$ of $\mathbb{A}$-open sets forms a topology.*

Proof. Closure of $\tau_{\mathbb{A}}$ under taking (arbitrary) unions follows from Theorem 172, together with the observation that

$$S \subseteq A \text{ is open with respect to } \mathbb{A} \text{ iff } \Delta_S \text{ is a bisimulation on } \mathbb{A}, \tag{46}$$

which in its turn is an immediate consequence of (42). We skip the proof of the fact that the *intersection* of two opens is open, since it requires a little more work. We refer the reader to Gumm & Schröder [54] for the details. ❑

It follows from the Proposition above that, given a subset S of (the carrier of) a coalgebra $\mathbb{A}$, there is a largest subcoalgebra of $\mathbb{A}$ (of which the carrier is) contained in S: Its universe is given as the union of all open subsets of S. It also follows from Proposition 182 that the collection $\tau_{\mathbb{A}}$ of open subsets of A forms a *complete* lattice under set inclusion. Hence, given a subset S of A, there is an open set $U \subseteq A$ which is the *meet* of the collection $\{Q \in \tau_{\mathbb{A}} \mid S \subseteq Q\}$. However, there is no guarantee that U is

also the *intersection* of this collection, or, indeed, that S is actually a subset of U. Thus we may not in general speak of the smallest subcoalgebra containing a given subset, as the following example from Gumm [50] witnesses.

EXAMPLE 183. Consider the standard Euclidean topology on the real numbers, seen as a coalgebra for the filter functor $\mathcal{F}_{\breve{\mathcal{P}}}$, cf. Example 144. One can show, that a set S of reals is open in the topological sense iff it is open in the sense of Definition 180 — in fact, this holds for any topology. Now take an arbitrary point r in $\mathbb{R}$. Obviously, we have that the *meet* of all open neighborhoods containing r is the empty set.

Before we turn to further coalgebraic constructions, consider the following natural link between homomorphic images and subcoalgebras.

PROPOSITION 184. *Given a coalgebraic homomorphism $\varphi : \mathbb{A} \to \mathbb{B}$, there is a (unique) subcoalgebra $\varphi[\mathbb{A}]$ of $\mathbb{B}$ such that $\varphi : \mathbb{A} \to \varphi[\mathbb{A}]$ is a surjective homomorphism.*

Proof. For a proof of this proposition, let $S := \varphi[A]$ be the (set-theoretic) image of A under φ, and let $f : S \to A$ be a right inverse of φ, that is, $\varphi(f(s)) = s$ for all $s \in S$. Now define $\sigma : S \to \Omega S$ by $\sigma := \Omega\varphi \circ \alpha \circ f$. It can be shown that the resulting structure $\mathbb{S}$ is always a subcoalgebra of $\mathbb{B}$, and that $\varphi : \mathbb{A} \to \mathbb{S}$ is a surjective homomorphism; for details the reader is referred to Rutten [97]. ❑

Our last example of a coalgebraic construction concerns the straightforward generalization of the disjoint union of Kripke models and frames. The idea is as follows. Recall that in Set, a concrete representation of the coproduct of a collection $\{A_i \mid i \in I\}$ of sets is given by the disjoint union $\biguplus_I A_i$, together with the inclusions/embeddings $e_i : A_i \to \biguplus_I A_i$. Hence, the defining property of coproducts provides the key ingredient of the coalgebraic notion of a coproduct, or sum of a family of coalgebras.

DEFINITION 185. The *sum* $\coprod_I \mathbb{A}_i$ of a family $\{\mathbb{A}_i \mid i \in I\}$ of coalgebras for some set functor Ω, is defined by endowing the disjoint union $A := \biguplus_I A_i$ with the unique map $\alpha : A \to \Omega A$ which turns all embeddings $e_i : A_i \to A$ into homomorphisms.

We have now gathered all the basic class operations needed to define the notion of a covariety, which was introduced in Rutten [97] as the natural dual of a variety in universal algebra.

DEFINITION 186. Let Ω be some endofunctor on Set. A class of Ω-coalgebras is a *covariety* if it closed under taking homomorphic images, subcoalgebras and sums. The smallest covariety containing a class K of Ω-coalgebras is called the covariety *generated* by K, notation: $\mathsf{Covar}(\mathsf{K})$.

As in the case of universal algebra, in order to obtain a more succinct characterization of the covariety generated by a class of coalgebras, one may develop a calculus of class operations.

DEFINITION 187. Let H, S and Σ denote the class operations of taking (isomorphic copies of) homomorphic images, subcoalgebras, and sums, respectively.

On the basis of these (and other) operations one may investigate the validity of 'inequalities' like $\mathsf{HS} \leq \mathsf{SH}$ (meaning that $\mathsf{HS}(\mathsf{K}) \subseteq \mathsf{SH}(\mathsf{K})$ for all classes K of coalgebras). Results of these kind lead to the following coalgebraic analog of Tarski's HSP-theorem in universal algebra, due to Gumm & Schröder [53].

THEOREM 188. *Let* K *be a class of* Ω*-coalgebras for some set functor* Ω*. Then*

$$\mathsf{Covar}(\mathsf{K}) = \mathsf{SH\Sigma}(\mathsf{K}).$$

Proof. It is straightforward to prove the theorem on the basis of the idempotency of the class operations H, S and Σ, together with the following three 'inequalities': $\mathsf{HS} \leq \mathsf{SH}$, $\Sigma\mathsf{S} \leq \mathsf{S}\Sigma$, and $\Sigma\mathsf{H} \leq \mathsf{H}\Sigma$. For proofs of these (and more) inequalities, the reader is referred to Gumm & Schröder [53]. ❑

As in the case of varieties, one may wonder about the basic building blocks of varieties. Dualizing the notion of subdirect irreducibility, we arrive at the following definition. It uses the notion of a conjunct sum, which is known, in the case of Kripke frames, under the name of *bounded union.*

DEFINITION 189. Let $\mathbb{A}$ be some Ω-coalgebra for some set functor Ω. A *conjunct representation* $\mathbb{A}$ by a family $\{\mathbb{A}_i \mid i \in I\}$ of coalgebras is a family of embeddings $\{e_i : \mathbb{A}_i \to \mathbb{A} \mid i \in I\}$ such that $A = \bigcup_{i \in I} e_i[A_i]$. In this case we call $\mathbb{A}$ a *conjunct sum* of the $\mathbb{A}_i$. A coalgebra $\mathbb{A}$ is called *conjunctly irreducible* if each of its conjunct representations is trivial in the sense that one of the embeddings is an isomorphism.

Covarieties are easily seen to be closed under taking conjunct sums — we will use this fact without further notice.

Given the results on dualizing the notion of subdirect irreducibility in section 5, in particular, Theorem 68, one would expect that conjunct irreducibility can be explained in terms of roots. Call a state s of a system $\mathbb{S}$ a *root* of $\mathbb{S}$ if $\mathbb{S}$ itself is the only subcoalgebra of $\mathbb{S}$ that contains s. It is then fairly easy to prove that a coalgebra is conjunctly irreducible if and only if it has a root. However, Gumm [50] proves that there is no analog of Birkhoff's s.i. theorem here, at least not for an arbitrary functor. For instance, expanding Example 183, one easily shows that a topological coalgebra will generally not be a conjunct sum of rooted coalgebras.

13 MODAL LOGIC AND COALGEBRAS

If coalgebras are mathematical structures that represent the essence of dynamics, then there is an obvious need for logics to represent and reason about properties of such structures. This is of particular importance for computer scientists who are interested in the formal specification and verification of the behavior of a system. The *kind* of properties that one wants to describe formally may differ from one application to another, but it seems natural to restrict attention to properties that are invariant under behavioral equivalence. Moss [11, 86] was the first to realize that such properties can be conveniently formalized in a version of *modal logic*, properly generalized from Kripke structures to systems for an arbitrary set functor. This connection between modal logic and coalgebra has provided a quite active research area. At the time of writing, quite a few proposals for coalgebraic modal logics are around; most of them are roughly based on one of the approaches to be discussed in this section.

We start with Moss' original approach, which is also the most general. In order to introduce his formalism, we first put ordinary modal logic in a slightly different perspective by introducing a new connective ∇. The *meaning* of this modality, which takes a

set of formulas as its argument, can be summarized by presenting the formula $\nabla\Phi$, with Φ a set of formulas, as the following abbreviation:

$$\nabla\Phi := \Box \bigvee \Phi \wedge \bigwedge \Diamond\Phi, \tag{47}$$

where $\Diamond\Phi$ denotes the set $\{\Diamond\varphi \mid \varphi \in \Phi\}$, and $\bigvee$ and $\bigwedge$ denote disjunction and conjunction. We do not want to exclude the possibility that Φ is an infinite set — coalgebraic logic is generally of an infinitary nature. The operator ∇ pops up in a number of areas in modal logic, cf. for instance the characteristic formulas of Chapter 5. We may also decide to treat this ∇ as a *primitive* connective. As long as we keep $\vee$ and $\top$ in our language, both the standard diamond and box connective are definable in terms of ∇, since we have the following equivalences:

$$\begin{array}{rcl} \Diamond\varphi & \equiv & \nabla\{\varphi, \top\}, \\ \Box\varphi & \equiv & \nabla\varnothing \vee \nabla\{\varphi\}, \end{array}$$

so that we may in fact *replace* the diamond and box with this new modality.

Spelling out the truth definition of $\nabla\Phi$, we see that it can in fact be expressed in terms of the *relation lifting* that we defined in section 11. For, let $\mathbb{S} = \langle S, \lambda, R[\cdot]\rangle$ be a modal model in coalgebraic shape. Then it is straightforward to verify that $\mathbb{S}, s \Vdash \nabla\Phi$ if and only if the pair $(R[s], \Phi)$ belongs to the relation lifting $\overline{\mathcal{P}}(\Vdash_{\mathbb{S}})$ of the satisfaction relation $\Vdash_{\mathbb{S}} \subseteq S \times \Phi$: Every $\varphi \in \Phi$ must hold at some successor $t \in R[s]$, and at every successor t of s some $\varphi \in \Phi$ must hold, see (43). This fundamental insight paves the way for Moss' development of *coalgebraic logic*, in which the same principle is applied to an arbitrary (but fixed) set functor Ω. Basically, the idea is to have

$$\mathbb{S}, s \Vdash_{\mathbb{S}} \nabla P \text{ iff } (P, \sigma(s)) \in \overline{\Omega}(\Vdash_{\mathbb{S}}).$$

Note that in this perspective, the satisfaction relation is much like a bisimulation between a language and a coalgebra; this observation was first made and exploited in Baltag [8].

In order to provide a more precise definition, recall from Remark 161 that we may uniquely extend Ω to a set based endofunctor Ω^+ on the category SET that has classes as objects, and set-continuous functions as arrows. For convenience, we follow Moss [86] in that we confine our attention to *standard* set functors, that is, functors that map inclusions to inclusions.

DEFINITION 190. Let $\Omega : \mathsf{Set} \to \mathsf{Set}$ be a standard set functor that preserves weak pullbacks. Then $\mathcal{L}_\Omega$, the language of *coalgebraic formulas for* Ω, is defined as the least class C such that (i) $\bigwedge \Phi \in \mathcal{L}_\Omega$ if $\Phi \subseteq \mathcal{L}_\Omega$ is a *set* of formulas, and (ii) $\nabla P \in \mathcal{L}_\Omega$ for any $P \in \Omega^+(\mathcal{L}_\Omega)$.

Categorically, $\langle \mathcal{L}_\Omega, \bigwedge, \nabla\rangle$ can be characterized as the *initial algebra* of the functor $(\mathcal{P} + \Omega)^+$. This explains our move to the category SET: if we want to guarantee the *existence* of such a structure, for reasons similar as given in the discussion following Proposition 158, we need to allow class-based algebras.

DEFINITION 191. Let $\Omega : \mathsf{Set} \to \mathsf{Set}$ preserve weak pullbacks. Given an Ω-coalgebra $\mathbb{S} = \langle S, \sigma\rangle$, define $\Vdash_{\mathbb{S}} \subseteq S \times \mathcal{L}_\Omega$ as the least relation satisfying

$$\begin{array}{rcl} s \Vdash_{\mathbb{S}} \bigwedge \Phi & \text{if} & s \Vdash_{\mathbb{S}} \varphi \text{ for all } \varphi \in \Phi, \\ s \Vdash_{\mathbb{S}} \nabla P & \text{if} & (P, \sigma(s)) \in \overline{\Omega}(W) \text{ for some set } W \subseteq \Vdash_{\mathbb{S}}. \end{array}$$

EXAMPLE 192. Consider the functor $\mathcal{P}\mathsf{Prop} \times \mathcal{P}$ of Kripke models. Unraveling the definitions, we find that an arbitrary element of $\Omega^+(\mathcal{L}_\Omega)$ must be of the form (A, Φ) with $A \subseteq \mathsf{Prop}$ a set of proposition letters, and $\Phi \subseteq \mathcal{L}_\Omega$ a set of formulas. It is not hard to verify that

$$\mathbb{S}, s \Vdash \nabla(A, \Phi) \text{ iff } \mathbb{S}, s \Vdash \bigwedge A \wedge \bigwedge \neg(\mathsf{Prop} \setminus A) \wedge \Box \bigvee \Phi \wedge \bigwedge \Diamond\Phi,$$

where $\bigwedge \neg(\mathsf{Prop} \setminus A)$ denotes the formula $\bigwedge\{\neg p \mid p \in P \setminus A\}$. It is instructive to observe the difference between this and (47) which displays an arbitrary ∇-formula for the functor $\mathcal{P}$ of Kripke frames as opposed to models.

EXAMPLE 193. For another example, an arbitrary element of the class $\Omega^+(\mathcal{L}_\Omega)$, where Ω is now the functor $\mathcal{I} \times \mathcal{I}$, must be a *pair* of formulas, say, (φ_0, φ_1). Clearly then we have

$$\mathbb{S}, s \Vdash \nabla(\varphi_0, \varphi_1) \text{ iff } \mathbb{S}, \pi_0(\sigma(s)) \Vdash \varphi_0 \text{ and } \mathbb{S}, \pi_1(\sigma(s)) \Vdash \varphi_1.$$

This in fact implies that *all* formulas are true at *all* states of *all* coalgebras; in other words, in the absence of propositions, the language $\mathcal{L}_\Omega$ may be rather uninteresting.

Obviously, many variations of this language exist, or may be defined. For instance, it is easy to develop finitary versions of the language, while independently of this, one may add Boolean connectives like negation or (infinitary) disjunction. Interestingly, $\mathcal{L}_\Omega$ on its own is already powerful enough to characterize behavior. Theorem 194 below shows that it has the *Hennessy-Milner property* (cf. Chapter 5 of this volume): non-bisimilarity of two points is witnessed by some formula in the language.

THEOREM 194. *Let $\Omega : \mathsf{Set} \to \mathsf{Set}$ preserves weak pullbacks, and let $\mathbb{S}$ and $\mathbb{S}'$ be two Ω-coalgebras. Then for any pair of states $s \in S$, $s' \in S'$:*

$$\mathbb{S}, s \leftrightarroweq \mathbb{S}', s' \textit{ iff } s \textit{ and } s' \textit{ satisfy the same } \mathcal{L}_F\textit{-formulas.}$$

Proof. The direction from left to right is proved by induction on the complexity of formulas. That is, we define Θ to be the class of formulas on which all bisimilar points in $\mathbb{S}$ and $\mathbb{S}'$ agree. Then we prove that $\Theta = \mathcal{L}_\Omega$ by showing that Θ is closed under $\bigwedge$ and ∇ (in the sense that $\bigwedge \Phi \in \Theta$ for all subsets $\Phi \subseteq \Theta$, and that $\nabla P \in \Theta$ for all $P \in \Omega(\Theta)$). We leave the fairly straightforward details as an exercise for the reader.

The proof for the other direction is analogous to that of Karp's Theorem for modal logic (see Chapter 5 of this volume), so we confine ourselves to a brief sketch here. Given an Ω-system $\mathbb{S}$, by ordinal induction we define a family $\varphi^{\mathbb{S}}_\alpha : S \to \mathcal{L}_\Omega$ as follows (we omit the superscript):

$$\begin{array}{rcl}\varphi_0(s) & := & \top \\ \varphi_{\alpha+1}(s) & := & \nabla(\Omega\varphi_\alpha)(\sigma(s)), \\ \varphi_\lambda(s) & := & \bigwedge\{\varphi_\alpha(s) \mid \alpha < \lambda\}.\end{array}$$

One approach to the proof would then be to show that the relation $\equiv_\varphi$, defined via $s \equiv_\varphi t$ if $\varphi_\alpha(s) = \varphi_\alpha(t)$ for all α, is itself a bisimulation. ❑

Moss' definition provides powerful languages, of which syntax and semantics uniformly depend on the coalgebraic signature, but his systems are not very welcoming to our intuitions on modal languages as extensions of propositional logic with diamonds and boxes

that are interpreted via accessibility relations. Baltag [8] introduces variants of Moss' language in which the connectives $\Box \bigvee \Phi$ and $\bigwedge \Diamond \Phi$ of (47) are (separately) generalized from Kripke frames to arbitrary functors, but also his formalism is far too abstract for practical purposes. It therefore seems worthwhile to develop more 'concrete' and practical alternatives to $\mathcal{L}_\Omega$.

In the case of Kripke polynomial functors, the concrete, inductive definition of the functor allows for more down to earth modal languages, as was first observed by Kurz [77]. Here we present a formalism that was introduced in Rößiger [95], and studied by Jacobs [62]. From the perspective of modal logic, its only non-standard feature is that both its syntax and semantics are *sorted* by the set $IngK$ of ingredient functors of K.

DEFINITION 195. Fix a Kripke polynomial functor K. We define the language $Fma_K = \bigcup_{\Lambda \in Ing(K)} Fma_K(\Lambda)$ of *K-sorted modal formulas*, by the following induction. (All functors appearing in the definition below are supposed to be ingredient functors of K.)

- $\bot \in Fma_K(\Lambda)$ for every $\Lambda \in Ing(K)$;
- if $\varphi, \psi \in Fma_K(\Lambda)$ then $\neg\varphi, \varphi \vee \psi \in Fma_K(\Lambda)$;
- if $c \in C$ then $c \in Fma_K(C)$;
- if $\varphi \in Fma_K(\Lambda_i)$ then $\Diamond_{\kappa_i}\varphi \in Fma_K(\Lambda_0 + \Lambda_1)$;
- if $\varphi \in Fma_K(\Lambda_i)$ then $\Diamond_{\pi_i}\varphi \in Fma_K(\Lambda_0 \times \Lambda_1)$;
- if $\varphi \in Fma_K(\Lambda)$ then $\Diamond_d\varphi \in Fma_K(\Lambda^D)$ for all $d \in D$;
- if $\varphi \in Fma_K(\Lambda)$ then $\Diamond_\ni\varphi \in Fma_K(\mathcal{P}\Lambda)$;
- if $\varphi \in Fma_K(K)$ then $\odot\varphi \in Fma_K(\mathcal{I})$.

We say that φ is *of sort* Λ if $\varphi \in Fma_K(\Lambda)$ — note that this sort need not be unique.

How do we interpret these formulas in coalgebras? Intuitively, with each K-coalgebra $\mathbb{S}$, we associate a multi-sorted frame based on the set $\bigcup_{\Lambda \in Ing(K)} \Lambda(S)$. The accessibility relations of this frame (which we will not make explicit) are completely determined by the shape of the functor. For instance, to link the set $(\Lambda_0 + \Lambda_1)(S)$ to $\Lambda_0(S)$, we lay down the relation $R_{\kappa_0} = \{(\kappa_0 s_0, s_0) \mid s_0 \in \Lambda_0(S)\}$. Likewise, the converse membership relation $\ni$ provides the accessibility relation from $\mathcal{P}\Lambda(S)$ to $\Lambda(S)$.

DEFINITION 196. Let $\mathbb{S} = \langle S, \sigma \rangle$ be a K-coalgebra for some Kripke polynomial functor K. By formula induction we define a sorted satisfaction relation $\Vdash = \bigcup_{\Lambda \in Ing(K)} \Vdash_\Lambda$,

with $\Vdash_\Lambda \subseteq \Lambda(S) \times Fma_K(\Lambda)$:

$$\begin{array}{rcl}
s \Vdash_\Lambda \bot & : & \text{never}, \\
s \Vdash_\Lambda \neg\varphi & \text{if} & s \not\Vdash_\Lambda \varphi \text{ (but } s \in \Lambda(S)), \\
s \Vdash_\Lambda \varphi \vee \psi & \text{if} & s \Vdash_\Lambda \varphi \text{ or } s \Vdash_\Lambda \psi, \\
s \Vdash_C c & \text{if} & s = c, \\
s \Vdash_{\Lambda_0+\Lambda_1} \Diamond_{\kappa_i}\varphi & \text{if} & s = \kappa_i(t) \text{ for some } t \in \Lambda_i(S) \text{ with } t \Vdash_{\Lambda_i} \varphi, \\
s \Vdash_{\Lambda_0\times\Lambda_1} \Diamond_{\pi_i}\varphi & \text{if} & s = (s_0, s_1) \text{ and } s_i \Vdash_{\Lambda_i} \varphi, \\
s \Vdash_{\Lambda^D} \Diamond_d\varphi & \text{if} & s(d) \Vdash_\Lambda \varphi, \\
s \Vdash_{\mathcal{P}\Lambda} \Diamond_\ni\varphi & \text{if} & \text{there is some } t \in s \text{ with } t \Vdash_\Lambda \varphi, \\
s \Vdash_{\mathcal{I}} \odot\varphi & \text{if} & \sigma(s) \Vdash_K \varphi.
\end{array}$$

Furthermore we employ the usual terminology concerning validity, etc.

EXAMPLE 197. Consider the functor $\Omega = \mathcal{P}\mathsf{Prop} \times \mathcal{P}(\mathcal{I} \times \mathcal{I})$ corresponding to Kripke models based on frames with a ternary accessibility relation T. In the standard modal language for such models, we would be working with a binary modality $\Diamond$, whereas here, we are dealing with four unary modalities: $\odot$, $\Diamond_\ni$, $\Diamond_{\pi_1}$ and $\Diamond_{\pi_2}$. We leave it for the reader to verify that the modal formula $\varphi_1\Diamond\varphi_2$ in the first language can be rendered as $\odot\Diamond_\ni(\Diamond_{\pi_0}\varphi_1 \wedge \Diamond_{\pi_1}\varphi_2)$ in the second. That is, we have

$$\mathbb{S}, s \models \odot\Diamond_\ni(\Diamond_{\pi_0}\varphi_1 \wedge \Diamond_{\pi_1}\varphi_2) \text{ iff there are } t_1, t_2 \text{ with } Tst_1t_2 \text{ and } \mathbb{S}, t_i \Vdash \varphi_i.$$

Bisimulation invariance of this language is easily proved:

PROPOSITION 198. *Assume that K is some Kripke polynomial functor, and let $\mathbb{S}$ and $\mathbb{S}'$ be two K-coalgebras. Then for any pair of states $s \in S$, $s' \in S'$:*

$$\mathbb{S}, s \leftrightarroweq \mathbb{S}', s' \textit{ only if } s \textit{ and } s' \textit{ satisfy the same formulas in } Fma_K.$$

Proof. Fix a bisimulation B between $\mathbb{S}$ and $\mathbb{S}'$. We claim that for any formula φ of type $\Lambda \in Ing(K)$, it holds for any pair $(s, s') \in \Lambda(S) \times \Lambda(S')$ that

$$\mathbb{S}, s \Vdash_\Lambda \varphi \text{ iff } \mathbb{S}', s' \Vdash_\Lambda \varphi,$$

provided that (s, s') belong to the relation lifting $\overline{\Lambda}(B)$ of B. The proof is by a straightforward formula induction. ❑

The basic modal theory of this formalism has been developed. For instance, analogous to Theorem 38 in Chapter 5 of this volume, one may prove that if K is a *finitary* Kripke polynomial functor, then the language Fma_K has the Hennessy-Milner property. Also, results concerning completeness and decidability are known. The interested reader is referred to Rößiger [95] and Jacobs [62].

We now move to the third approach towards coalgebraic modal logic. Pattinson [89] combines the generality of the first formalism with the concreteness of the second. That is, the approach applies to arbitrary set functors, but provides languages with standard diamonds and boxes. First we present a simplified version, which is based on the idea

to *extract diamonds out of the natural transformations* from the coalgebra functor Ω to the power set functor $\mathcal{P}$. Recall that a natural transformation $\lambda : \Omega \to \mathcal{P}$ provides an arrow $\lambda_S : \Omega(S) \to \mathcal{P}(S)$ for each set S, in such a way that for each function $f : S \to S'$, the following diagram commutes:

$$\begin{tikzcd} S \arrow[d, "f"'] & \Omega S \arrow[r, "\lambda_S"] \arrow[d, "\Omega f"'] & \mathcal{P}S \arrow[d, "\mathcal{P}f"'] \\ S' & \Omega S' \arrow[r, "\lambda_{S'}"] & \mathcal{P}S' \end{tikzcd}$$

Thus if we have an Ω-coalgebra $\mathbb{S} = \langle S, \sigma \rangle$, we may define a relation $R_\lambda \subseteq S \times S$ for such a λ by putting $R_\lambda st$ if $t \in \lambda_S(\sigma(s))$. We may then introduce a diamond $\Diamond_\lambda$ which takes this R_λ as its accessibility relation. Natural transformations $\lambda : \Omega \to \mathcal{P}$ thus literally transform Ω-coalgebras into $\mathcal{P}$-coalgebras, that is, Kripke frames.

Similarly, if we want to have *atomic propositions* in our language, consider any natural transformation ν from Ω to the constant functor Prop. We then make $p \in \mathsf{Prop}$ true at s depending on whether p is an element of the set $\nu_S(\sigma(s))$ or not. It is as if we add the valuation V_ν to $\mathbb{S}$ given by $V_\nu(p) := \{s \in \mathbb{S} \mid p \in \nu_S(\sigma(s))\}$.

DEFINITION 199. Let $\Omega : \mathsf{Set} \to \mathsf{Set}$ be some functor, $\nu : \Omega \to \mathsf{Prop}$ some natural transformation, and Λ some collection of natural transformations $\Omega \to \mathcal{P}$. Then $\mathcal{L}_{\nu,\Lambda}$ is the standard modal language we obtain by taking Prop as the collection of propositional variables, and $\tau_\Lambda := \{\Diamond_\lambda \mid \lambda \in \Lambda\}$ as the modal similarity type.

It will now be obvious how these formulas are interpreted in Ω-coalgebras. We confine ourselves to the following clauses of the inductive truth definition:

$$\begin{array}{rll} \mathbb{S}, s \Vdash p & \text{if} & p \in \nu(\sigma(s)), \\ \mathbb{S}, s \Vdash \Diamond_\lambda \varphi & \text{if} & \mathbb{S}, t \Vdash \varphi \text{ for some } t \in \lambda_S(\sigma(s)). \end{array}$$

In other words, an Ω-coalgebra $\mathbb{S}$ is treated as the Kripke model $\langle S, \{R_\lambda \mid \lambda \in \Lambda\}, V_\nu \rangle$. The reason to require the transformations to be *natural* is to guarantee invariance under behavioral equivalence.

PROPOSITION 200. *Let Ω, ν and Λ be as in Definition 199. Then for any pair $\mathbb{S}$, $\mathbb{S}'$ of Ω-coalgebras, and any pair of states $s \in S$, $s' \in S'$:*

$$\mathbb{S}, s \equiv_\Omega \mathbb{S}', s' \textit{ only if } s \textit{ and } s' \textit{ satisfy the some } \mathcal{L}_{\nu,\Lambda}\textit{-formulas.}$$

Proof. It suffices to prove that for any coalgebraic homomorphism $f : \mathbb{S} \to \mathbb{S}'$, each state s in $\mathbb{S}$ satisfies the same $\mathcal{L}_{\nu,\Lambda}$-formulas as $f(s)$ in $\mathbb{S}'$. This inductive proof is in fact straightforward, the crucial observation being that the naturality of the transformations guarantees that f is a bounded morphism between the Kripke models associated with $\mathbb{S}$ and $\mathbb{S}'$. ❑

For the more general picture, Pattinson uses *predicate liftings* (from $\mathcal{P}S$ to $\mathcal{P}\Omega S$) to obtain modal operators. In order to introduce these, note that the semantics of the modal operator $\Diamond_\lambda$ could have been expressed as follows:

$$\mathbb{S}, s \Vdash \Diamond_\lambda \varphi \text{ iff } \sigma(s) \in \mu_S^\lambda([\![\varphi]\!]),$$

where $\mu^{\lambda}_S : \mathcal{P}S \to \mathcal{P}\Omega S$ is given by $A \mapsto \{\Gamma \in \Omega S \mid \lambda_S(\Gamma) \cap A \neq \varnothing\}$, and $[\![\varphi]\!]$ denotes the extension of φ in $\mathbb{S}$. In fact, it can be shown that μ^{λ} is a natural transformation from the contravariant power set functor $\breve{\mathcal{P}}$ to the functor $\breve{\mathcal{P}} \circ \Omega$. Generalizing this, we arrive at the following definition.

DEFINITION 201. A *predicate lifting* for a set functor Ω is a natural transformation $\mu : \breve{\mathcal{P}} \to \breve{\mathcal{P}} \circ \Omega$. With each predicate lifting we can associate a modal operator $\Diamond_{\mu}$, with the following semantics:

$$\mathbb{S}, s \Vdash \Diamond_{\lambda}\varphi \text{ iff } \sigma(s) \in \mu_S([\![\varphi]\!]).$$

And as before, it is the naturality of the transformation that ensures that this language is invariant under behavioral equivalence.

In order to finish this section, a number of remarks are in order. First, the above mentioned versions of coalgebraic logic are open for the standard expressive enhancements that we know from extended modal logic. As examples we mention Jacobs [64], who adds past operators (as in section 8.1) to a variant of the formalism defined in the Definitions 195 and 196, and Venema [107], who develops a finitary fixed point version of Moss' logic.

Second, it should be mentioned that for certain *polynomial* functors, coalgebraic specification languages have been developed of an *equational* rather than modal nature. Very roughly, the idea is that coalgebras for such a polynomial functor K can be represented by a structured collection of partial functions on the carrier of the coalgebra. From the perspective of Definition 196, this can be explained by the observation that in the absence of the power set functor, each and every accessibility relation of the multi-sorted frame is in fact (the graph of) a partial function. Lacking the space for an appropriate survey of this more equational perspective, we only mention one interesting idea which adds some *modal flavor* to equational logic. In coalgebraic approaches towards specification theory, such as that of *hidden algebra*, a *state equation* $t_1 \simeq t_2$ holds of a state s in a coalgebra $\mathbb{S}$ if $t_1^{\mathbb{S}}(s)$ and $t_2^{\mathbb{S}}(s)$ evaluate to *bisimilar* (rather than identical) states in $\mathbb{S}$. We refer the reader to Goldblatt [42, 43] and Roşu [94] for more details; in particular, Goldblatt [43] contains a clear discussion of this overlap area between modal and equational logic.

Third, Kurz & Pattinson [78] establish a link between coalgebraic predicates and the *final sequence*, see Remark 162: they argue that finitary predicates correspond to subsets of some set Z_n (n finite) occurring in the final sequence. This work is in fact closely related to that of Ghilardi [33], even though the word 'coalgebra' is not mentioned in the latter work.

Finally, there is an interesting connection between Hennessy-Milner results and final coalgebras: Goldblatt [45] proves that a set functor Ω admits a final coalgebra iff there is a coalgebraic modal language for Ω, which has the Hennessy-Milner property and is based on a *set* (rather than a proper class) of formulas.

14 CO-BIRKHOFF THEOREMS AND COFREE COALGEBRAS

In order to give the reader some impression of universal coalgebra at work, we discuss one result, or better, one cluster of results, in some detail. The topic that we have chosen concerns the coalgebraic version of Birkhoff's variety theorem; recall that this result in universal algebra states that a class C of algebras is a *variety*, (that is, closed under the

class operations H, S and P), if and only if it is equationally definable. Thus in essence, Birkhoff established a link between two different ways of characterizing algebraic classes: a logical one, in terms of the validity of certain formulas, and a structural one, in terms of certain class operations.

If we are after a co-Birkhoff result, two roads seem open to us. Since we have already developed the concept of a covariety, the most obvious thing to do would be to try and find out what corresponds to it, logically. An alternative approach would be to investigate the structural counterpart of the logical languages developed in the previous section. Here we follow the first road, but interestingly, it leads us to (very natural generalizations of) modal languages! This provides justification for our earlier claim that modal logic is dual to equational logic.

In the proof of Birkhoff's theorem, *free* algebras play a key role; thus it will come as no surprise that we will be looking at *cofree* coalgebras here. However, these structures do not serve as proof tools only, they have a quite intuitive meaning as well. To explain this, first note that many set functors provide coalgebraic structures that come with a notion of *output*. For instance, the black box machines of Example 141 may be prompted to display some value, the states of the automata of Example 142 output 0 or 1 depending on whether they are final or not, and the states of a Kripke model satisfy some set of propositional variables. For a general functor $\Omega : \mathsf{Set} \to \mathsf{Set}$, such a notion of output may not be available. However, nothing prevents us from *adding* an extra output feature to the functor.

DEFINITION 202. Let Ω be some set functor, and C a set of objects that we will call *colors*. A *C-coloring* of an Ω-coalgebra $\mathbb{A} = \langle A, \alpha \rangle$ is a map $\gamma : A \to C$; the structure $\langle A, \alpha, \gamma \rangle$ will be called the coalgebra $\mathbb{A}$ *colored* by γ.

As a prime example, Kripke models can be seen as $\mathcal{P}\mathsf{Prop}$-colored Kripke frames. In general, C-colored Ω-coalgebras may be identified with Ω_C-coalgebras, where Ω_C is the functor $C \times \Omega$; this provides us with a category of C-colored Ω-coalgebras. Spelling it out, $f : S \to S'$ is a morphism from $\langle S, \sigma, \gamma \rangle$ to $\langle S', \sigma', \gamma' \rangle$ if f is an Ω-coalgebra homomorphism from $\langle S, \sigma \rangle$ to $\langle S', \sigma' \rangle$ such that $\gamma(s) = \gamma'(fs)$ for all $s \in S$.

Colors can be seen as the coalgebraic duals of variables, colorings as the duals of assignments. This brings us to the definition of a cofree coalgebra, which is the formal dual of the notion of a free algebra. We recall the latter notion, for the purposes of the present context, as follows. Let $\Omega : \mathsf{Set} \to \mathsf{Set}$ be some set functor, X a set of variables, and $T = \langle T, \tau : \Omega T \to T \rangle$ some Ω-algebra such that $e : X \to T$ is some kind of injection. (Here we deviate from the more standard presentation, where e is taken to be an inclusion map.) Then T, with e, is called *free over* X if for every Ω-algebra $\mathbb{A} = \langle A, \alpha \rangle$ and every assignment $f : X \to A$, there is a unique homomorphism $\widetilde{f} : \mathbb{T} \to \mathbb{A}$ such that $f = \widetilde{f} \circ e$.

DEFINITION 203. Let Ω be a set functor, C a set of colors, and $\mathbb{Z}$ some Ω-coalgebra with a coloring $\gamma : Z \to C$. Then $\mathbb{Z}$ (with γ) is called *(absolutely) cofree over* C if for every Ω-coalgebra $\mathbb{A} = \langle A, \alpha \rangle$ and every coloring $g : A \to C$ of $\mathbb{A}$, there is a unique homomorphism $\widetilde{g} : \mathbb{A} \to \mathbb{Z}$ such that $g = \gamma \circ \widetilde{g}$.

$$\begin{tikzcd} & Z \arrow[d, "\gamma"] \\ A \arrow[ur, dotted, "\widetilde{g}"] \arrow[r, "g"'] & C \end{tikzcd}$$

Observe that the diagram above is not properly typed (it mixes arrows from different categories). A more proper formulation of the notion of cofreeness would involve the right adjoint to the forgetful functor from $\mathsf{Coalg}(\Omega)$ to Set.

It is immediate from the definitions that an Ω-coalgebra with coloring $\gamma : T \to C$ is

cofree over C iff the structure $\langle T, \tau, \gamma\rangle$ is a *final* coalgebra for the functor $\Omega_C = C \times \Omega$. This explains that we may view the carrier Z of such a cofree coalgebra as the collection of all *behavior patterns* expressible in the output set C. And this perspective paves the way for a dual version of Birkhoff's variety theorem, by providing a natural means for characterizing classes of coalgebras in terms of permitted, or forbidden, behaviors.

DEFINITION 204. Let Ω be some set functor, and let $\mathbb{Z}$, with coloring $\gamma : Z \to C$, be the cofree coalgebra over some set C of colors. Given a set Q in $\mathbb{Z}$, let $\mathsf{Cov}(Q)$ be the class of Ω-coalgebras $\mathbb{A}$ such that $\eta[A] \subseteq Q$ for all homomorphisms $\eta : \mathbb{A} \to \mathbb{Z}$.

And conversely, given a class K of Ω-coalgebras, define $Bhv(\mathsf{K}) \subseteq Z$ to be the union of all images $\widetilde{g}[A]$ in which $\widetilde{g}$ arises from some C-coloring g of some coalgebra $\mathbb{A}$ in K.

There are all kinds of interesting facts concerning these two maps. For instance, it is fairly obvious from the definitions that Bhv and Cov form a (dual) Galois connection: For any class K of Ω-coalgebras, and any set Q of behavior patterns, we have

$$Bhv(\mathsf{K}) \subseteq Q \text{ iff } \mathsf{K} \subseteq \mathsf{Cov}(Q). \tag{48}$$

We will have use for this fact in the proof of a first co-Birkhoff result, which is basically due to Rutten [97]. In the remainder of this section we restrict our attention to *small* functors, in order to ensure the existence of final and cofree coalgebras.

THEOREM 205. *Let Ω be some endofunctor on* Set *which is κ-small for some cardinal κ. Then for any set C of size κ, the cofree coalgebra over C exists, and a class* K *of Ω-coalgebras is a covariety iff* $\mathsf{K} = \mathsf{Cov}(Q)$ *for some set Q of behavior patterns.*

Proof. It follows from the assumption on Ω that the functor $\Omega_C = C \times \Omega$ has a final coalgebra. However, we already observed that this structure may be represented as a triple $\langle Z, \zeta, \gamma\rangle$ such that $\mathbb{Z} = \langle Z, \zeta\rangle$, with coloring γ, is the cofree Ω-coalgebra over C. We fix this $\mathbb{Z}$ and γ for the remainder of the proof.

In order to show that $\mathsf{Cov}(Q)$ is a covariety, one needs to subsequently prove closure under taking homomorphic images, subcoalgebras, and sums. Here we restrict our attention to the proof for subcoalgebras, because that is the only part where the cofreeness of $\mathbb{Z}$ is used.

Suppose that $\mathbb{A}$ is a subcoalgebra of $\mathbb{B}$, with inclusion ι, while $\mathbb{B}$ belongs to $\mathsf{Cov}(Q)$; we need to show that $\mathbb{A}$ also belongs to this class. For that purpose, consider a homomorphism $\eta : \mathbb{A} \to \mathbb{Z}$, and observe that $\gamma \circ \eta : A \to C$ is a coloring of $\mathbb{A}$. Clearly this coloring can be extended to a coloring $g : B \to C$ of $\mathbb{B}$. Let $\widetilde{g} : \mathbb{B} \to \mathbb{Z}$ be the unique homomorphism such that $g = \gamma \circ \widetilde{g}$ — such a map exists by the cofreeness of $\mathbb{Z}$.

$$\begin{tikzcd}
A \arrow[r, "\eta"] \arrow[d, "\iota"'] & Z \arrow[d, "\gamma"] \\
B \arrow[ur, "\widetilde{g}"] \arrow[r, "g"] & C
\end{tikzcd}$$

Now $g = \gamma \circ \widetilde{g}$, so that $\gamma \circ \widetilde{g} \circ \iota = g \circ \iota$. But g was chosen so that $g \circ \iota = \gamma \circ \eta$. Hence we find that $\gamma \circ \widetilde{g} \circ \iota = \gamma \circ \eta$, so by the cofreeness of $\mathbb{Z}$ with respect to colorings of $\mathbb{A}$, we find that $\widetilde{g} \circ \iota = \eta$, that is, $\widetilde{g}$ extends η. From this it is immediate that $\eta[A] = \widetilde{g}{\restriction_A}[A] \subseteq \widetilde{g}[B]$, so that $\eta[A] \subseteq Q$ by the assumption that $\mathbb{B}$ belongs to $\mathsf{Cov}(Q)$.

For the other direction of the theorem, suppose that K is a covariety; we claim that

$$\mathsf{K} = \mathsf{Cov}(Bhv(\mathsf{K})). \tag{49}$$

The inclusion $\subseteq$ is immediate from (48). For the opposite inclusion, it easily follows from the definitions that $Bhv(\mathsf{K})$ is $\mathbb{Z}$-open. Let $\mathbb{B}_{\mathsf{K}}$ be the (unique) subcoalgebra of $\mathbb{Z}$ with

carrier set $Bhv(\mathsf{K})$. It is not hard to prove that $\mathbb{B}_{\mathsf{K}}$ is a conjunct sum of algebras in K, which implies that $\mathbb{B}_{\mathsf{K}}$ actually belongs to K since covarieties are closed under taking conjunct sums. Hence, in order to prove the remaining inclusion $\supseteq$ of (49), it suffices to show that

$$\text{every coalgebra in } \mathsf{Cov}(Bhv(\mathsf{K})) \text{ is a conjunct sum of subcoalgebras of } \mathbb{B}_{\mathsf{K}}. \tag{50}$$

Take an arbitrary coalgebra $\mathbb{A}$ in $\mathsf{Cov}(Bhv(\mathsf{K}))$. From the κ-smallness of Ω it may be derived that $\mathbb{A}$ is the conjunct sum of coalgebras $\mathbb{A}_i$, each of size at most κ. Clearly then it suffices to prove that each $\mathbb{A}_i$ belongs to K, since covarieties are closed under taking conjunct sums.

Fix some $i \in I$; clearly $\mathsf{Cov}(Bhv(\mathsf{K}))$, being closed under taking subcoalgebras, contains $\mathbb{A}_i$. Since $|A_i| \leq \kappa = |C|$, there is an injective coloring $e_i : A_i \to C$. Hence by cofreeness of $\mathbb{Z}$ there is a unique homomorphism $\widetilde{e}_i : \mathbb{A}_i \to \mathbb{Z}$ such that $e_i = \gamma \circ \widetilde{e}_i$. This $\widetilde{e}_i$ must also be injective, which implies that $\mathbb{A}_i$ is isomorphic to its image $\widetilde{e}_i[\mathbb{A}_i]$. But, since $\mathbb{A}_i$ belongs to $\mathsf{Cov}(Bhv(\mathsf{K}))$, the structure $\widetilde{e}_i[\mathbb{A}_i]$ is a subcoalgebra of $\mathbb{B}_{\mathsf{K}}$, and thus, belongs to K. From this it is immediate that each $\mathbb{A}_i$ belongs to K, and thus, so does the conjunct sum $\mathbb{A}$. ❑

Clearly, not only the statement, but also the proof of Theorem 205 is dual to that of Birkhoff's variety theorem. For instance, the coalgebra $\mathbb{B}_{\mathsf{K}}$ clearly fulfills the role of the *cofree* coalgebra *for the class* K over the color set C. What seems to be missing from Theorem 205, however, is some notion of logic, involving *syntax*. (It should be noted that also in the algebraic case, the straightforward characterization of varieties in terms of equations only obtains in the case of relatively simple functors.) Since we are discussing a dual of Birkhoff's theorem, the question this raises is: what are *co-equations*?

Given the nature of systems as state-based models of dynamics, it seems natural to require that formulas describe *behavior*. This would provide natural constraints on possible coequational languages, namely, that formulas are evaluated at states, in such a way that truth is invariant under behavioral equivalence. Furthermore, we allow the use of colors in order to obtain sufficient expressive power. It was an insight of Kurz [76] that these requirements may also be read as a natural *definition* of coalgebraic modal logic.

DEFINITION 206. Let Ω be some set functor. A *coalgebraic modal language* for Ω consists of a set C of colors, a class $\mathcal{L}_C$ of formulas, and, for each C-colored Ω-coalgebra $\langle \mathbb{S}, g\rangle$, a truth or satisfaction relation $\Vdash^{\mathbb{S},g} \subseteq S \times \mathcal{L}_C$ such that $\Vdash$ is invariant under behavioral equivalence. That is, if $\langle \mathbb{S}, g\rangle, s \equiv_{\Omega_C} \langle \mathbb{T}, h\rangle, t$, then $\mathbb{S}, s \equiv_{\mathcal{L}_C} \mathbb{T}, t$, where the latter notation indicates that s in $\langle \mathbb{S}, g\rangle$ and t in $\langle \mathbb{T}, h\rangle$ satisfy exactly the same $\mathcal{L}_C$-formulas φ.

In the sequel we will use notation and terminology from modal logic. For instance, we write $\langle \mathbb{S}, g\rangle, s \Vdash \varphi$ instead of $s \Vdash^{\mathbb{S},g} \varphi$, and we define $\mathbb{S}, g \Vdash \varphi$ and $\mathbb{S} \Vdash \varphi$ by quantifying over all elements and all valuations, respectively.

How can we link such modal languages to the cofree coalgebra? The idea here is that modal formulas correspond to subcoalgebras: if $\mathbb{Z}$, with C-coloring γ is a cofree coalgebra over C, then define

$$[\![\varphi]\!]^{\mathbb{Z},\gamma} := \{z \in Z \mid \mathbb{Z}, \gamma, z \Vdash \varphi\}.$$

Using the behavioral invariance of the logic, it is not hard to see that $[\![\varphi]\!]$ (we usually omit superscripts) is always $\mathbb{Z}$-open. Now one way to obtain nice co-Birkhoff results is to require the modal language to be expressive enough for the converse to hold as well.

DEFINITION 207. Let Ω be some κ-small set functor, and let $\langle C, \mathcal{L}_C, \Vdash\rangle$, with $|C| = \kappa$ constitute a coalgebraic modal logic for Ω. This modal logic is called *expressive* if every open set of the C-cofree coalgebra $\mathbb{Z}$ is of the form $[\![\varphi]\!]$ for some formula φ.

This may seem a strong requirement on a language, but expressive languages are not hard to come by.

EXAMPLE 208. Under some mild additional assumptions on Ω, one may show that Moss' logic of Definition 190 and 191, extended with infinite *disjunctions*, is expressive. For a proof sketch: strengthen Theorem 194 by proving that for any pointed Ω_C-system $(\mathbb{S}, s)$, there is a formula $\varphi^{\mathbb{S},s}$ such that for all pointed Ω_C-system $(\mathbb{S}', s')$ one has that $\mathbb{S}', s' \Vdash \varphi^{\mathbb{S},s}$ iff $\mathbb{S}', s' \equiv_{\Omega_C} \mathbb{S}, s$. Then, given an open set U of the cofree Ω_C-coalgebra $\mathbb{Z}$, one may define $\varphi^U := \bigvee\{\varphi^{\mathbb{Z},u} \mid u \in U\}$.

Now the next theorem bears witness to the tight link between modal logic and coalgebras. It is due to Kurz [76], while a very similar result was proved in Gumm & Schröder [53].

THEOREM 209. *Let Ω be some C-small set functor, and let $\langle C, \mathcal{L}_C, \Vdash\rangle$ constitute an expressive coalgebraic modal logic for Ω. Then a class K of Ω-coalgebras is a covariety iff for some formula φ, K is the class of all Ω-coalgebras $\mathbb{S}$ such that $\mathbb{S} \Vdash \varphi$.*

Proof. Let $\mathbb{Z}$, with coloring $\gamma : Z \to C$, be the cofree Ω-coalgebra over C. Given a formula φ, it is a direct consequence of cofreeness and truth invariance, that for any Ω-coalgebra $\mathbb{S}$ with C-coloring g, and for any state s in $\mathbb{S}$, we have

$$\mathbb{S}, g, s \Vdash \varphi \text{ iff } \widetilde{g}(s) \in [\![\varphi]\!], \tag{51}$$

from which one easily derives that for any $\mathcal{L}_C$-formula φ:

$$\mathsf{Cov}([\![\varphi]\!]) \text{ is the class of all } \Omega\text{-coalgebras } \mathbb{S} \text{ such that } \mathbb{S} \Vdash \varphi. \tag{52}$$

From (52) the direction '$\Leftarrow$' of the Theorem is immediate. For the other direction, suppose that K is a covariety. Then by expressiveness, $Bhv(\mathsf{K}) = [\![\varphi]\!]$ for some formula φ, so by (49) and (52) it follows that $\mathsf{K} = \mathsf{Cov}(Bhv(\mathsf{K})) = \mathsf{Cov}([\![\varphi]\!])$, as required. ❑

Although this theorem, being formulated in terms of a fairly general notion of modal logic, may still seem to be rather abstract, it does provide a useful tool to provide more concrete results. For instance, given Example 208, as a corollary to Theorem 209 one may obtain very general modal co-Birkhoff results for Moss' coalgebraic logic. Or, to give an even more concrete corollary of Theorem 209, call an (ordinary) modal frame *κ-bounded* for some cardinal κ if every point has less than κ successors.

COROLLARY 210. *A class K of κ-bounded frame is (within the class of all κ-bounded frames) definable by means of infinitary modal formulas, if and only if K is closed under taking generated subframes, homomorphic images and disjoint unions.*

The reader who compares the above two result to the Goldblatt-Thomasson Theorem 79, may be puzzled by the absence of ultrafilter extensions here. The explanation

for this absence is of course that such Stone-type completions are not relevant in the presence of infinite disjunctions and conjunctions. If one takes the alternative road to co-Birkhoff theorems and starts, not from the notion of a covariety, but from a *finitary* coalgebraic logical formalism, one will find that notions like ultrafilter extensions or ultraproducts are needed in the characterization of definable classes of coalgebras. Results in this direction can be found in for instance Goldblatt [42, 43] or Roşu [94].

Finally, the search for coalgebraic versions of Birkhoff's variety theorem has received considerable attention in the coalgebraic literature, as is witnessed by many contributions in [66, 93, 20, 52, 4]. Perhaps Gumm [51] should get some special mentioning for developing an alternative coequational syntax based on equivalence classes of infinite labeled trees.

15 DUALITY OF ALGEBRA AND COALGEBRA

Various other coalgebraic topics may be of interest to modal logicians, but here we confine ourselves to a brief discussion of the duality between algebra and coalgebra.

In remark 152 we already observed that some of the similarities between algebra and coalgebra are based on the fact that a coalgebra $\mathbb{C} = \langle C, \gamma : C \to \Omega C \rangle$ over an endofunctor $\Omega : \mathsf{C} \to \mathsf{C}$ can also be seen as an algebra in the *opposite* category C^{op}. In fact, it is a trivial exercise to show that

$$\mathsf{Coalg}(\Omega) = (\mathsf{Alg}(\Omega^{op}))^{op}. \tag{53}$$

That is, the category of Ω-coalgebras is dually isomorphic to the category of algebras over the functor Ω^{op} (which acts on objects and arrows just like Ω does, the difference being that Ω^{op} is an endofunctor on C^{op}).

This duality between algebras and coalgebras has been a major guideline in the development of universal coalgebra, see Rutten [97]. To mention just one example (many more can be found in the text): whereas *initial* algebras play an important role in universal algebra, it is the *final* objects that are relevant in coalgebra. For instance, whereas the principle of induction is based on the fact that initial algebras have no proper subalgebras, the dual *coinduction* principle boils down to the fact that final coalgebras have no proper quotients. However, it is important to realize that in (53) the *base* category has been dualized. This means, for instance, that systems, or Set-coalgebras, correspond, not so much to algebras over Set, as to algebras over the opposite category Set^{op} (which happens to be equivalent to the category of complete and atomic Boolean algebras with complete homomorphisms). As a consequence, a general theory of systems cannot be obtained by a straightforward dualization of universal (Set-based) algebra. On the other hand, the fact that systems are, just like standard algebras, 'sets with structure', indicates that many universal algebraic concepts may apply to coalgebra by *analogy* rather than by duality — see for instance Proposition 179. Thus, the universal coalgebraic theory of systems is an interesting mix of dualized and non-dualized universal algebra, with, of course, some characteristics of its own.

In case that there is an informative *duality* for the base category C, more can be said of (53). This applies for instance to the just mentioned duality of the category Set, but for the present purpose we prefer to focus on the category Stone of Stone spaces. The point is, that since Stone is dually equivalent to the well-known category BA of Boolean

algebras, every endofunctor Ω on Stone induces an endofunctor $\Omega^* := (\cdot)_* \circ \Omega \circ (\cdot)^*$ on BA. It is then an immediate consequence of (53) that the categories $\mathsf{Coalg}(\Omega)$ and $\mathsf{Alg}(\Omega^*)$ are dually equivalent:

$$\mathsf{Coalg}(\Omega) \rightleftharpoons \mathsf{Alg}(\Omega^*). \tag{54}$$

For an example of this, consider the Vietoris functor $\mathcal{V}$ of Example 147. Concretely, the behavior of its dual functor $\mathcal{V}^* : \mathsf{BA} \to \mathsf{BA}$ on objects is as follows. To a Boolean algebra $\mathbb{B}$ it assigns the Boolean algebra $\mathcal{V}^*(\mathbb{B})$ freely generated by the set $\{\Diamond b \mid b \in B\}$, subject to the axioms $\Diamond\bot = \bot$ and $\Diamond a \vee \Diamond b = \Diamond(a \vee b)$. Since the category $\mathsf{Coalg}(\mathcal{V})$ is dually equivalent to that of modal algebras, we thus see that the latter category, MA, may be represented as an algebraic category $\mathsf{Alg}(\mathcal{V}^*)$. This insight in fact provided the very first connection between modal logic and coalgebra, see Abramsky [1]. Recently, the duality that (54) provides between algebra and coalgebra has been used to prove results on coalgebraic modal *logics*, where we now use the word 'logic' in the technical sense. For instance, Jacobs [62] and Kupke, Kurz & Venema [74] use dualities in the style of (54) to prove completeness results for the multi-sorted modal logic of Definition 195 and 196. Kupke, Kurz & Pattinson [73] apply the above framework in order to characterize properties of arbitrary coalgebraic modal logics.

Let us finish the chapter with the observation that *both* of the fundamental dualities underlying the mathematical theory of modal logic are nontrivial instances of an algebra/coalgebra duality. This means that the algebraic and the coalgebraic approach towards modal logic may be fruitfully operated in tandem. We believe that a thorough study of the interaction of algebra and coalgebra will provide a better understanding, not only of modal logic itself, but also of its mathematical surroundings.

ACKNOWLEDGEMENTS

Many people have contributed to this chapter by commenting on earlier versions of it. I am grateful to Nick Bezhanishvili, Mai Gehrke, Çiğdem Gencer, Rob Goldblatt, H. Peter Gumm, Gaelle Fontaine, Helle Hansen, Ian Hodkinson, Ramon Jansana, Clemens Kupke, Alessandra Palmigiano, Jan Rutten, and Mark Theunissen for their help. Great is my gratitude to Alexander Kurz, who provided a large number of suggestions to improve the chapter.

A BASICS OF UNIVERSAL ALGEBRA AND CATEGORY THEORY

This section provides some technical preliminaries to this Chapter; we briefly review notation and terminology on universal algebra and category theory.

If we equip a set with a collection of finitary operations, we call the resulting structure an *algebra*; two such structures are called *similar* if their operations correspond in number and rank. In order to formalize this notion we introduce the notion of a *similarity type* as a set Σ of function symbols each of which comes with a nonnegative integer to be called its *rank* or *arity*. The arity of a function symbol f is denoted as $ar(f)$. Function symbols of rank zero are called *constants*.

The similarity type of (bounded) lattices is the set $Latt = \{\top, \bot, \wedge, \vee\}$ where $\top$ ('top') and $\bot$ ('bottom') are constants, and $\wedge$ ('meet') and $\vee$ ('join') are binary symbols. As

the similarity type for Boolean algebras we take the set $Bool = \{\top, \bot, \neg, \wedge, \vee\}$ where $\top$, $\bot$, $\wedge$ and $\vee$ are as before, and $\neg$ ('complementation') is a unary symbol.

A Σ-algebra is then a pair $\mathbb{A} = (A, I)$, in which the *interpretation* I assigns to each function symbol $f \in \Sigma$ an operation of arity $ar(f)$ on the *carrier* A of the algebra. Usually we write $f^{\mathbb{A}}$ rather than $I(f)$, and denote the algebra $\mathbb{A} = \langle A, I\rangle$ by $\mathbb{A} = \langle A, \{f^{\mathbb{A}} \mid f \in \Sigma\}\rangle$. As an example, let, for a set S, $\mathbb{P}(S) = \langle \mathcal{P}(S), S, \varnothing, \sim_S, \cap, \cup\rangle$ be the power set algebra, where $\sim_S$ denotes the unary operation of complementation with respect to S. An algebra is called *trivial* if it has just one element; this completely determines the behavior of the operations.

A *homomorphism* from a Σ-algebra $\mathbb{A}$ to a similar algebra $\mathbb{B}$ is a map $\theta : A \to B$ that preserves Σ-structure, in the sense that, for all $f \in \Sigma$, and all $a_1, \ldots, a_n$ in A (where $n = ar(f)$):

$$\theta(f^{\mathbb{A}}(a_1, \ldots, a_n)) = f^{\mathbb{B}}(\theta a_1, \ldots, \theta a_n). \tag{55}$$

An injective homomorphism is called an *embedding* and a surjective one, an *epimorphism*; an *isomorphism* is a bijective homomorphism. A homomorphism with the same source as target algebra is called an *endomorphism* in general, and an *automorphisms* if it is bijective.

Homomorphisms are closely related to special equivalence relations: a *congruence* on $\mathbb{A}$ is an equivalence relation $\sim$ satisfying, for all $f \in \Sigma$:

$$\text{if } a_1 \sim b_1 \;\&\; \ldots \;\&\; a_n \sim b_n, \text{ then } f^{\mathbb{A}}(a_1, \ldots, a_n) \sim f^{\mathbb{A}}(b_1, \ldots, b_n), \tag{56}$$

where n is the rank of f. Given a congruence $\sim$ on $\mathbb{A}$, the *quotient algebra of* $\mathbb{A}$ *by* $\sim$ is the algebra $\mathbb{A}/{\sim}$ whose carrier is the set $A/{\sim} := \{[a] \mid a \in A\}$ of equivalence classes of A under $\sim$, and whose operations are defined by

$$f_{\mathbb{A}/\sim}([a_1], \ldots, [a_n]) = [f_{\mathbb{A}}(a_1, \ldots, a_n)].$$

(This is well-defined by (56).) The close connection between homomorphisms and congruences is formed by the fact that if $\theta : \mathbb{A} \to \mathbb{B}$ is a homomorphism, its *kernel* $\ker(\theta) := \{(a, b) \in A \times A \mid \theta(a) = \theta(b)\}$ is a congruence on $\mathbb{A}$, while, on the other hand, for any congruence $\sim$ on $\mathbb{A}$, the associated *natural map* $\nu_\sim$ taking an element $a \in A$ to its equivalence class $[a]$ is a surjective homomorphism from $\mathbb{A}$ onto $\mathbb{A}/{\sim}$.

The set of congruences $Cg\mathbb{A}$ of an algebra $\mathbb{A}$ forms in fact a complete lattice under the subset ordering; this lattice is denoted as $\mathbb{C}\mathsf{g}(\mathbb{A})$; the meet operation of this lattice is simply their intersection, while the join of two congruences is given by $\Theta_1 \vee \Theta_2 = \Theta_1 \cup (\Theta_1 \circ \Theta_2) \cup (\Theta_1 \circ \Theta_2 \circ \Theta_1) \cup \cdots$.

A Σ-algebra $\mathbb{A}$ is a *subalgebra* of a Σ-algebra $\mathbb{B}$ if $A \subseteq B$ and for all $f \in \Sigma$, the operation $f^{\mathbb{A}}$ coincides with the restriction of $f^{\mathbb{B}}$ to A. The *direct product* $\mathbb{A} = \prod_{i \in I} \mathbb{A}_i$ of a family of Σ-algebras is an algebra with carrier $\prod_{i \in I} A_i$ and such that for $f \in \Sigma$ and $a_1, \ldots, a_n \in \prod_{i \in I} A_i$:

$$f^{\mathbb{A}}(a_1, \ldots, a_n)(i) := f^{\mathbb{A}_i}(a_1(i), \ldots, a_n(i))$$

We assume familiarity with the notions of ultraproduct and ultrapower.

Given a class K of algebras, we let $\mathsf{H}(\mathsf{K})$ denote the class of homomorphic images of algebras in K; $\mathsf{S}(\mathsf{K})$ is the class of isomorphic copies of subalgebras of algebras in K, and likewise definitions applies for the class operations P (products), Pu (ultraproducts) and Pw (ultrapowers).

A class of algebras is called a *variety* if it is closed under taking subalgebras, homomorphisms, and products; the smallest variety containing a class K is called the variety *generated by* K, notation: $\mathsf{Var}(\mathsf{K})$. Using inequalities like $\mathsf{SH} \leq \mathsf{HS}$ (meaning that, for any class of algebras K, $\mathsf{SH}(\mathsf{K})$ is a subclass of $\mathsf{HS}(\mathsf{K})$), together with the idempotence of the class operations S, H and P, one can prove Tarski's Theorem stating that

$$\mathsf{Var}(\mathsf{K}) = \mathsf{HSP}(\mathsf{K}) \tag{57}$$

for any class of algebras K.

Given a similarity type Σ and a set of variables X, we define the set $Ter_\Sigma(X)$ of Σ-*terms over* X by a straightforward induction: it is the smallest including X which contains $f(t_1, \ldots, t_n)$ whenever it contains $t_1, \ldots, t_n$ and $f \in \Sigma$ is a function symbol of rank n. (In particular, $Ter_\Sigma(X)$ contains all constants in Σ.) In this chapter we adopt the convention that unless explicitly indicated otherwise, X denotes a countably infinite set of variables; we often omit explicit reference to X, writing for instance Ter_Σ rather than $Ter_\Sigma(X)$, etc. Also, writing $s(x_1, \ldots, x_n)$ for a term s, we indicate that the variables occurring in s are among $x_1, \ldots, x_n$.

Given an assignment α of a set X of variables to (the carrier A of) an algebra $\mathbb{A}$, we inductively define the *meaning* $\widetilde{\alpha}(s)$ of a term s as follows:

$$\begin{array}{lcl} \widetilde{\alpha}(x) & = & \alpha(x) \\ \widetilde{\alpha}(f(t_1, \ldots, t_n)) & = & f^{\mathbb{A}}(\widetilde{\alpha}(t_1), \ldots, \widetilde{\alpha}(t_n)). \end{array} \tag{58}$$

Thus any term $s(x_1, \ldots, x_n)$ induces an n-ary *term function* $s^{\mathbb{A}}$ on $\mathbb{A}$, given by

$$s^{\mathbb{A}}(a_1, \ldots, a_n) = \widetilde{\alpha}(s),$$

where α is any assignment mapping each x_i to a_i. (Of course, $s^{\mathbb{A}}$ can also be given an inductive definition.)

Using the close resemblance between the second clause of (58) and (55), we can turn the meaning function into a real homomorphism by imposing Σ-algebra structure on the set $Ter_\Sigma(X)$, obtaining the *term algebra* $\mathbb{T}\mathsf{er}_\Sigma(X)$. The idea is to interpret the function symbol $f \in \Sigma$ as follows:

$$f^{\mathbb{T}\mathsf{er}_\Sigma(X)} : (t_1, \ldots, t_n) \mapsto f(t_1, \ldots, t_n).$$

Elaborating on this perspective, let K be a class of Σ-algebras, and $\mathbb{F}$ a Σ-algebra generated by a set $X \subseteq F$. Suppose that for every $\mathbb{A}$ in K and every map $\alpha : X \to A$ there is a homomorphism $\widetilde{\alpha} : \mathbb{F} \to \mathbb{A}$ extending α. Then we say that $\mathbb{F}$ has the *universal mapping property for* K *over* X, or that $\mathbb{F}$ is *free for* K *over* X. The identities of (58) thus reveal that $\mathbb{T}\mathsf{er}_\Sigma(X)$ is free over X for the class of *all* Σ-algebras; for this reason it is often referred to as the *absolutely free algebra over* X.

Free algebras have a number of important properties of which we mention the following:

- every algebra in K is a homomorphic image of a free algebra over an appropriately large set of generators;
- all free algebras for K belongs to the class $\mathsf{SP}(\mathsf{K})$;
- if $\mathbb{F}$ and $\mathbb{F}'$ are free for K over the generator sets X and X', respectively, and X and X' have the same cardinality, then $\mathbb{F}$ and $\mathbb{F}'$ are isomorphic.

Universal algebra may on the one hand be seen as generalizing the study of individual classes of algebras such as groups, fields, or lattices. On the other hand we may consider it as a rather special branch of model theory in which one is interested in structures for a language without relation symbols. The standard language for talking about such structures is *equational*.

An *equation* is nothing but a pair (s,t) of terms, always denoted as $s \approx t$. The equation $s \approx t$ (with $s,t \in Ter_\Sigma(X)$) *is true* or *holds* in the algebra $\mathbb{A}$ *under the assignment* $\alpha : X \to A$, notation: $\mathbb{A} \models_\alpha s \approx t$ if s and t obtain the same meaning in $\mathbb{A}$ under α, that is, if $\widetilde{\alpha}(s) = \widetilde{\alpha}(t)$. An equation $s \approx t$ *holds in* the algebra $\mathbb{A}$, or, equivalently, the algebra $\mathbb{A}$ *satisfies* the equation $s \approx t$, notation: $\mathbb{A} \models s \approx t$, if $\mathbb{A} \models_\alpha s \approx t$ for every assignment α.

The relation $\models$ induces a Galois connection between sets of formulas and classes of algebras; the polarities of this connection are given as the maps Equ and Mod, where $Equ(\mathsf{K})$ is the set of all equations that hold in K, and $\mathsf{Mod}(E)$ denotes the class of algebras that satisfy every equation in E. The classes of algebras that are stable under this connection, that is, the classes K of the form $\mathsf{Mod}(E)$ for some set E of equations, are called *equational classes*. An important result by Birkhoff states that this notion coincides with that of a variety, and that for any class K of algebras it holds that

$$\mathsf{Mod}(Equ(\mathsf{K})) = \mathsf{Var}(\mathsf{K}). \tag{59}$$

The relation

$$s \equiv_{\mathsf{K}} t \;:\Longleftrightarrow\; \mathsf{K} \models s \approx t$$

corresponding to the set $Equ(\mathsf{K})$ is in fact a *congruence* on the term algebra $\mathbb{T}\mathsf{er}_\Sigma$. The algebra $\mathbb{T}\mathsf{er}_\Sigma(X)/\equiv_{\mathsf{K}}$ has the universal mapping property for K over $[X]$ (the set of equivalence classes of X under $\equiv_{\mathsf{K}}$), which, together with the third fact on free algebras listed above, explains why we call it the *free algebra for* K over $[X]$.

A *category* C consists of a class $\mathsf{Ob}(\mathsf{C})$ of *objects*, and for each pair of objects A, B, a family $\mathsf{C}(A,B)$ of *arrows*. If f belongs to the latter set, we write $f : A \to B$, and call A the *domain* and B the *codomain* of the arrow. The collection of arrows is endowed with some algebraic structure: for every object A of C there is an arrow $id_A : A \to A$, and every pair $f : A \to B$, $g : B \to C$ can be uniquely *composed* to an arrow $g \circ f : A \to C$. These operations are supposed to satisfy the associative law for composition, while the appropriate identity arrows are left- and right neutral elements. An arrow $f : A \to B$ is an *iso* if it has an *inverse*, that is, an arrow $g : B \to A$ such that $f \circ g = id_B$ and $g \circ f = id_A$. Examples of categories are Set, the class of sets with functions, and, for every similarity type Σ, the class $\mathsf{Alg}(\Sigma)$ of Σ-algebras, with homomorphisms as arrows. The *opposite* category C^{op} of a given category C has the same objects as C, while $\mathsf{C}^{op}(A,B) = \mathsf{C}(B,A)$ for all objects A, B from C, and the operations on arrows are defined in the obvious way.

An object X is *initial* in a category C if for every object A in C there is a unique arrow $\alpha : X \to A$, and *final* if for all A there is a unique $\alpha : A \to X$. In Set, the empty set is initial, and the final objects are precisely the singletons. A *product* of two objects A_0 and A_1 in a category C consists of a triple $(A, \alpha_0 : A \to A_0, \alpha_1 : A \to A_1)$, such that for every triple $(A', \alpha_0' : A' \to A_0, \alpha_1' : A' \to A_1)$ there is a unique arrow $f : A' \to A$ such that $\alpha_i \circ f = \alpha_i'$ for both i. *Coproducts* of A_0 and A_1 are defined dually as triples $(A, \alpha_0 : A_0 \to A, \alpha_1 : A_1 \to A)$, such that for every triple $(A', \alpha_0' : A_0 \to A', \alpha_1' : A_1 \to A')$ there

is a unique arrow $f : A \to A'$ such that $f \circ \alpha_i = \alpha'_i$ for each i. The category Set *has* both products and coproducts — that is, every pair (S_0, S_1) of sets has both a product (for which we may take the cartesian product $S_0 \times S_1$ together with the two projection functions $\pi_i : S_0 \times S_1 \to S_i$), and a coproduct (for which we may take the disjoint union $S_0 \uplus S_1 = S_0 \times \{0\} \cup S_1 \times \{1\}$ together with the coproduct maps κ_0 and κ_1 given by $\kappa_i(s) = (s, i)$).

A *functor* $\Omega : \mathsf{C} \to \mathsf{D}$ from a category C to a category D consists of an operation mapping objects and arrows of C to objects and arrows of D, respectively, in such a way that $\Omega f : \Omega A \to \Omega B$ if $f : A \to B$, $\Omega(id_A) = id_{\Omega A}$ and $\Omega(g \circ f) = (\Omega g) \circ (\Omega f)$ for all objects and arrows involved. A functor $\Omega : \mathsf{C} \to \mathsf{D}^{op}$ is sometimes called a *contravariant* functor from C to D. An *endofunctor* on C is a functor $\Omega : \mathsf{C} \to \mathsf{C}$.

As examples we consider the following *set functors* (that is, endofunctors on Set): (i) for a fixed set C, the *constant* functor mapping all sets to C and all arrows to id_C; this functor is denoted as C, (ii) the *power set functor* $\mathcal{P}$, which maps any set S to its power set $\mathcal{P}S$, and any map $f : S \to S'$ to the map $\mathcal{P}f : \mathcal{P}S \to \mathcal{P}S'$ given by $\mathcal{P}f : X \mapsto \{fx \mid x \in X\}$, and (iii) for every cardinal κ, the variant $\mathcal{P}_\kappa$ of the power set functor, which maps any set S to the the collection $\mathcal{P}_\kappa S := \{X \subseteq S \mid \kappa > |X|\}$, and agrees with $\mathcal{P}$ on the arrows for which is defined. Furthermore, given two functors Ω_0 and Ω_1, their *product functor* $\Omega_0 \times \Omega_1$ is given (on objects) by $(\Omega_0 \times \Omega_1)S := \Omega_0 S \times \Omega_1 S$, while for $f : S \to S'$, the map $(\Omega_0 \times \Omega_1)f$ is given as $((\Omega_0 \times \Omega_1)f)(\sigma_0, \sigma_1) := ((\Omega_0 f)(\sigma_0), (\Omega_1 f)(\sigma_1))$. The *coproduct functor* is defined similarly. Finally, every category C admits the *identity functor* $\mathcal{I}_\mathsf{C} : \mathsf{C} \to \mathsf{C}$ which is the identity on both objects and arrows of C.

Let C and D be two categories, and let Ω and Ψ be two functors from C to D. A *natural transformation* τ from Ω to Ψ, notation $\tau : \Omega \Rightarrow \Psi$, consists of D-arrows $\tau_A : \Omega A \to \Psi A$ such that $\tau_B \circ \Omega f = \Psi f \circ \tau_A$ for each $f : A \to B$ in C.

Finally, let $\Omega : \mathsf{C} \to \mathsf{D}$ and $\Psi : \mathsf{D} \to \mathsf{C}$ be two functors linking the categories C and D. Ω and Ψ constitute an *equivalence* between C and D if their compositions are naturally isomorphic to the identity functors, that is, if there are natural transformations $\sigma : \mathcal{I}_\mathsf{C} \Rightarrow \Psi\Omega$ and $\tau : \mathcal{I}_D \Rightarrow \Omega\Psi$ such that all arrows $\sigma_A : A \to \Psi\Omega A$ and $\tau_B : B \to \Omega\Psi B$ are isos. If such Ω and Ψ exist, then the categories C and D are called *equivalent*; if Ω and Ψ are in fact each other's inverse (both on maps and on arrows) then C and D are *isomorphic*. If Ω and Ψ form a *dual equivalence* between the categories C and D, that is, an equivalence between the categories C and D^{op}, then we say that the categories are *dual* or *dually equivalent* to each other.

BIBLIOGRAPHY

[1] S. Abramsky. A Cook's Tour of the Finitary Non-Well-Founded Sets, 1988. Invited Lecture at BCTCS.

[2] P. Aczel. *Non-well-founded sets*, volume 14 of *CSLI Lecture Notes*. CSLI, 1988.

[3] P. Aczel and N. Mendler. A final coalgebra theorem. In D.H. Pitt, A. Poigné, and D.E. Rydeheard, editors, *Category Theory and Computer Science*, volume 389 of *Lecture Notes in Computer Science*. Springer, 1989.

[4] J. Adámek, editor. *Coalgebraic Methods in Computer Science (CMCS'04)*, volume ?? of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2004.

[5] J. Adámek and V. Koubek. On the greatest fixed point of a set functor. *Theoretical Computer Science*, 150:57–75, 1995.

[6] J. Adámek and H.-E. Porst. From varieties of algebras to covarieties of coalgebras. In Corradini et al. [20].

[7] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217–274, 1998.

[8] A. Baltag. A logic for coalgebraic simulation. In Reichel [93].

[9] M. Barr. Terminal coalgebras in well-founded set theory. *Theoretical Computer Science*, 114(2):299–315, 1993. (Corrigendum in *Theor.Comp.Sci.* 124:189–192,1994).

[10] M. Barr. Relational algebras. In S. Mac Lane, editor, *Reports of the Midwest Category Seminar IV*, volume 137 of *Lecture Notes in Mathematics*. Springer Verlag, 1997.

[11] J. Barwise and L. Moss. *Vicious Circles*, volume 60 of *CSLI Lecture Notes*. CSLI Publications, 1996.

[12] G. Birkhoff. *Lattice Theory*. Number 25 in Colloquium Publications. American Mathematical Society, Providence, 1967. Third edition.

[13] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Number 53 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001.

[14] W. Blok. *Varieties of Interior Algebras*. PhD thesis, Mathematisch Instituut, University of Amsterdam, 1976.

[15] W. Blok. The lattice of modal algebras: An algebraic investigation. *Journal of Symbolic Logic*, 45:221–236, 1980.

[16] W. Blok and D. Pigozzi. Algebraizable logics. *Memoirs of the American Mathematical Society*, 77(396), 1989.

[17] S. Burris and H.P. Sankappanavar. *A Course in Universal Algebra*. Graduate Texts in Mathematics. Springer, 1981.

[18] W. Buszkowski. Embedding Boolean structures into atomic Boolean structures. *Zeitschrift für mathemathische Logik und Grundlagen der Mathematik*, 32:227–228, 1986.

[19] A. Carboni, G. Kelly, and R. Wood. A 2-categorical approach to change of base and geometric morphisms I. Technical Report 90-1, Department of Pure Mathematics, University of Sydney, 1990.

[20] A. Corradini, M. Lenisa, and U. Montanari, editors. *Coalgebraic Methods in Computer Science (CMCS'01)*, volume 44 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2001.

[21] J. Czelakowski. *Protoalgebraic Logic*. Trends in Logic. Kluwer Academic Press, 2001.

[22] Paul Erdős. Graph theory and probability. *Canadian Journal of Mathematics*, 11:34–38, 1959.

[23] L. Esakia. Topological Kripke models. *Soviet Mathematics Doklady*, 15:147–151, 1974.

[24] L. Esakia. To the theory of modal and superintuitionistic systems. In V. Smirnov, editor, *Proceedings of the USSR Symposium on Logical Inference*, pages 147–172. Nauka, Moscow, 1979. (In Russian).

[25] K. Fine. Some connections between elementary and modal logic. In Kanger [71].

[26] J.M. Font and R. Jansana. *A general algebraic semantics for sentential logics*. Lecture Notes in Logic. Springer, 1996.

[27] M. Gehrke. The order structure of Stone spaces and the T_D-separation axiom. *Zeitschrift für mathemathische Logik und Grundlagen der Mathematik*, 37:5–15, 1991.

[28] M. Gehrke and J. Harding. Bounded lattice expansions. *Journal of Algebra*, 238:345–371, 2001.

[29] M. Gehrke, J. Harding, and Y. Venema. MacNeille completions and canonical extensions. *Transactions of the American Mathematical Society*, to appear.

[30] M. Gehrke and B. Jónsson. Bounded distributive lattices with operators. *Mathematica Japonica*, 40:207–215, 1994.

[31] M. Gehrke and B. Jónsson. Monotone bounded distributive lattices expansions. *Mathematica Japonica*, 52:197–213, 2000.

[32] M. Gehrke and B. Jónsson. Bounded distributive lattice expansions. *Mathematica Scandinavica*, 94:3–45, 2004.

[33] S. Ghilardi. An algebraic theory of normal forms. *Annals of Pure and Applied Logic*, 71:189–245, 1995.

[34] S. Ghilardi and G. Meloni. Constructive canonicity in non-classical logics. *Annals of Pure and Applied Logic*, 86:1–32, 1997.

[35] S. Givant. Universal classes of simple relation algebras. *The Journal of Symbolic Logic*, 64:575–589, 1999.

[36] S. Givant and Y. Venema. The preservations of Sahlqvist equations in completions of Boolean algebras with operators. *Algebra Universalis*, 41:47–84, 1999.

[37] R. Goldblatt. Metamathematics of modal logic I. *Reports on Mathematical Logic*, 6:41–78, 1976.

[38] R. Goldblatt. Metamathematics of modal logic II. *Reports on Mathematical Logic*, 7:21–52, 1976.

[39] R. Goldblatt. Varieties of complex algebras. *Annals of Pure and Applied Logic*, 38:173–241, 1989.

[40] R. Goldblatt. Elementary generation and canonicity for varieties of Boolean algebras with operators. *Algebra Universalis*, 34:551–607, 1995.

[41] R. Goldblatt. Algebraic polymodal logic: a survey. *Logic Journal of the IGPL*, 8:393–450, 2000.

[42] R. Goldblatt. A calculus of terms for coalgebras of polynomial functors. In Corradini et al. [20].

[43] R. Goldblatt. What is the coalgebraic analogue of Birkhoff's variety theorem. *Theoretical Computer Science*, 266:853–886, 2001.

[44] R. Goldblatt. Mathematical modal logic: a view of its evolution. *Journal of Applied Logic*, 1:309—392, 2003. An updated version is available from the web site of the author.

[45] R. Goldblatt. Final coalgebras and the hennessy-milner property. *Annals of Pure and Applied Logic*, to appear.

[46] R. Goldblatt, I. Hodkinson, and Y. Venema. Erdős graphs resolve Fine's canonicity problem. *Bulletin of Symbolic Logic*, 10:186–208, 2004.

[47] R. Goldblatt and S. Thomason. Axiomatic classes in propositional modal logic. In J. Crossley, editor, *Algebra and Logic*, pages 163–173. Springer, 1974.

[48] V. Goranko and S. Passy. Using the universal modality: Gains and questions. *Journal of Logic and Computation*, 2:5–30, 1992.

[49] V. Goranko and D. Vakarelov. Sahlqvist formulas unleashed in polyadic modal languages. In F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev, editors, *Advances in Modal Logic, Volume 3*, pages 221–240. World Scientific, 2002.

[50] H. Peter Gumm. Elements of the general theory of coalgebras, 1999. Lecture Notes for LUATCS'99, Rand Africaans University, Johannesburg; available from `http://www.mathematik.uni-marburg.de/~gumm/`.

[51] H. Peter Gumm. Birkhoffs variety theorem for coalgebras. *Contributions to General Algebra*, 13:159–173, 2000.

[52] H. Peter Gumm, editor. *Coalgebraic Methods in Computer Science (CMCS'03)*, volume 82 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2003.

[53] H. Peter Gumm and T. Schröder. Covarieties and complete covarieties. *Theoretical Computer Science*, 260:71–86, 2001.

[54] H. Peter Gumm and T. Schröder. Coalgebras of bounded type. *Mathematical Structures in Computer Science*, 12:565–578, 2002.

[55] H. Hansen and C. Kupke. A coalgebraic perspective on monotone modal logic. In Adámek [4].

[56] J. Harding. Canonical completions of lattices and ortholattices. *Tatra Mountains Mathematical Publications*, 15:85–96, 1998.

[57] L. Henkin, J.D. Monk, and A. Tarski. *Cylindric Algebras. Part 1. Part 2.* North-Holland Publishing Company, Amsterdam, 1971, 1985.

[58] R. Hirsch and I.M. Hodkinson. *Relation Algebras by Games*. Number 147 in Studies in Logic. Elsevier, Amsterdam, 2002.

[59] E. Hoogland. *Definability and Interpolation. Model-theoretic investigations*. PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam, 2001.

[60] T. Ihringer. *Allgemeine Algebra. Mit einem Anhang über Universelle Coalgebra von H. P. Gumm*, volume 10 of *Berliner Studienreihe zur Mathematik*. Heldermann Verlag, 2003.

[61] B. Jacobs. Objects and classes, coalgebraically. In B. Freitag, C. Jones, C. Lengauer, and H.-J. Schek, editors, *Object-Orientation with Parallelism and Persistence*, volume 44, pages 83–103. Kluwer Academic Publishers, 1997.

[62] B. Jacobs. Many-sorted coalgebraic modal logic: a model-theoretic study. *Theoretical Informatics and Applications*, 35(1):31–59, 2001.

[63] B. Jacobs. Exercises in coalgebraic specification. In R. Backhouse, R. Crole, and J. Gibbons, editors, *Algebraic and Coalgebraic Methods in the Mathematics of Program Construction*, volume 44 of *Lecture Notes in Computer Science*, pages 237–280. Springer, 2002.

[64] B. Jacobs. The temporal logic of coalgebras via Galois algebras. *Mathematical Structure in Computer Science*, 12:875–903, 2002.

[65] B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction. *Bulletin of the European Association for Theoretical Computer Science*, 62:222–259, 1997.

[66] B. Jacobs and J. Rutten, editors. *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 19 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 1999.

[67] P. Jipsen. Discriminator varieties of Boolean algebras with residuated operators. In C. Rauszer, editor, *Algebraic Methods in Logic and Computer Science*, volume 28 of *Banach Center Publications*, pages 239–252. Polish Academy of Sciences, 1993.

[68] P.J. Johnstone. *Stone Spaces*, volume 3 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1982.

[69] B. Jónsson. On the canonicity of Sahlqvist identities. *Studia Logica*, 4:473–491, 1994.

[70] B. Jónsson and A. Tarski. Boolean algebras with operators, Part I. *American Journal of Mathematics*, 73:891–939, 1952.

[71] S. Kanger, editor. *Proceedings of the Third Scandinavian Logic Symposium. Uppsala 1973*. North-Holland Publishing Company, 1975.

[72] M. Kracht. *Tools and Techniques in Modal Logic*. Number 142 in Studies in Logic. Elsevier, Amsterdam, 1999.

[73] C. Kupke, A. Kurz, and D. Pattinson. Algebraic semantics for coalgebraic logics. In Adámek [4].

[74] C. Kupke, A. Kurz, and Y. Venema. Stone coalgebras. *Theoretical Computer Science*, 327:109–134, 2004.

[75] A. Kurz. Coalgebras and modal logic, 1999. Lecture Notes for ESSLLI'01; available from `http://www.folli.uva.nl/CD/2001/courses/readers/K40.pdf`.

[76] A. Kurz. A co-variety-theorem for modal logic. In M. Zakharyaschev, K. Segerberg, M. de Rijke, and H. Wansing, editors, *Advances in Modal Logic, Volume 2*. CSLI Publications, 2000.

[77] A. Kurz. Specifying coalgebras with modal logic. *Theoretical Computer Science*, 260:119–138, 2001.

[78] A. Kurz and D. Pattinson. Coalgebraic modal logic of finite rank. *Mathematical Structures in Computer Science*, to appear. Also available as CWI Technical Report SEN-R0222, 2002.

[79] J. Lambek. A fixpoint theorem for complete categories. *Mathematische Zeitschrift*, 103:151–161, 1968.

[80] E. Lemmon. Algebraic semantics for modal logics, Parts I & II. *Journal of Symbolic Logic*, pages 46–65 & 191–218, 1966.

[81] T. Litak. On notions of completeness weaker than Kripke incompleteness. In R. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing, editors, *Advances in Modal Logic, Volume 5*. King's College London Publications, to appear.

[82] H. MacColl. *Symbolic Logic and its Applications*. Longmans, Green, and Co., London, 1906.

[83] L. Maksimova. Interpolation theorems in modal logic and amalgable varieties of topological Boolean algebras. *Algebra and Logic*, 18:348–370, 1979.

[84] M. Marx and Y. Venema. *Multidimensional Modal Logic*, volume 4 of *Applied Logic Series*. Kluwer Academic Publishers, 1997.

[85] J. Monk. Completions of Boolean algebras with operators. *Mathematische Nachrichten*, 46:47–55, 1970.

[86] L. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96:277–317, 1999. (Erratum published *Ann.P.Appl.Log.* 99:241–259, 1999).

[87] I. Németi. Cylindric-relativized set algebras have strong amalgamation. *Journal of Symbolic Logic*, 50:689–700, 1985.

[88] D. Pattinson. Semantical principles in the modal logic of coalgebras. In H. Reichel and A. Ferreira, editors, *Proc. 18th International Symposium on Theoretical Aspects of Computer Science (STACS 2001)*, volume 2010 of *LNCS*, Berlin, 2001. Springer.

[89] D. Pattinson. Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. *Theoretical Computer Science*, 309:177–193, 2003.

[90] D. Pattinson. An introduction to the theory of coalgebras, 2003. Notes accompanying a course at NASSLLI 2003; available from `http://www.indiana.edu/~nasslli/`.

[91] W. Rautenberg. *Klassische und nichtklassische Aussagenlogik*. Vieweg & Sohn, 1979.

[92] H. Reichel. An approach to object semantics based on terminal coalgebras. *Mathematical Structures in Computer Science*, 5:129–152, 1995.

[93] H. Reichel, editor. *Coalgebraic Methods in Computer Science (CMCS'00)*, volume 33 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2000.

[94] G. Roşu. Equational axiomatizability for coalgebra. *Theoretical Computer Science*, 260:229–247, 2001.

[95] M. Rößiger. Coalgebras and modal logic. In Reichel [93], pages 299–320.

[96] J. Rutten. Automata and coinduction (an exercise in coalgebra). In D. Sangiorigi and R. de Simone, editors, *Proceedings of CONCUR '98*, volume 1466 of *Lecture Notes in Computer Science*, pages 194–218. Springer, 1998.

[97] J. Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249:3–80, 2000.

[98] H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logic. In Kanger [71], pages 110–143.

[99] G. Sambin. Subdirectly irreducible modal algebras and initial frames. *Studia Logica*, 62:269–282, 1999.

[100] G. Sambin and V. Vaccaro. A topological proof of Sahlqvist's theorem. *Journal of Symbolic Logic*, 54:992–999, 1989.

[101] M. Stone. The theory of representations for Boolean algebras. *Transactions of the American Mathematical Society*, 40:37–111, 1936.

[102] S. Thomason. Semantic analysis of tense logics. *Journal of Symbolic Logic*, 37:150–158, 1972.

[103] S. Thomason. Categories of frames for modal logics. *Journal of Symbolic Logic*, 40:439–442, 1975.

[104] V. Trnková. Relational automata in a category and theory of languages. In M. Karpinski, editor, *Fundamentals of computation theory: proceedings of the 1977 international FCT-conference*, volume 56 of *Lecture Notes in Computer Science*, pages 340–355. Springer, 1977.

[105] V. Trnková. General theory of relational automata. *Fundamenta Informaticae*, 3:189–234, 1979.

[106] Y. Venema. Atomless varieties. *Journal of Symbolic Logic*, 68:607–614, 2003.

[107] Y. Venema. Automata and fixed point logics for coalgebras. In Adámek [4].

[108] Y. Venema. A dual characterization of subdirectly irreducible BAOs. *Studia Logica*, 77:105–115, 2004.

[109] J. Worrell. Terminal sequences for accessible endofunctors. In Jacobs and Rutten [66], pages 39–53.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

7

MODAL DECISION PROBLEMS

Frank Wolter and Michael Zakharyaschev

1	Modal Logic as 'die Klassentheorie'	427
2	Thomason's analysis	429
3	Normal modal logics	432
3.1	Blok's dichotomy	434
3.2	Chagrov's classification	439
3.3	Postmortem	444
4	Syntactical classes of modal logics	445
4.1	Sahlqvist logics	446
4.2	Uniform logics	448
4.3	Logics with $\Box\Diamond$-axioms	450
4.4	Logics with noniterative axioms	451
4.5	Modal reduction principles	451
4.6	Logics with n-variable axioms	452
5	Semantically constrained classes of modal logics	453
6	Frame-theoretic characterisation	457
6.1	Canonical formulas for **K4**	458
6.2	Canonical formulas for tense logics of linear time flows	464
7	Decision problems for tense logics	467
8	Subframe logics	470
9	Superintuitionistic logics	476
9.1	Intuitionistic frames	476
9.2	Canonical formulas	478
9.3	Modal companions and preservation theorems	480
9.4	Completeness	481
9.5	Medvedev's logic	482

1 MODAL LOGIC AS 'DIE KLASSENTHEORIE'

There are different views on the subject of Modal Logic. For the purpose of this chapter it is important to distinguish between two of them.

According to the *local* view, Modal Logic deals with a number of *concrete modal logics*. Since the beginning of the 20th century developers and users of Modal Logic from philosophy, mathematics, computer science, artificial intelligence, linguistics and other fields have introduced and investigated dozens of particular modal logics suitable for their

needs: *epistemic*, *provability*, *temporal*, *dynamic*, *description*, *spatial*, to mention just a few.

With the number of concrete modal logics introduced in the literature growing, there came an understanding that it may be interesting and important to formulate general abstract notions of modal logics and to investigate the landscape of the resulting classes of logics and their properties. The pioneers of this *global* approach were Scroggs [127] who considered all extensions of **S5**, Dummett and Lemmon [33] who studied all logics between **S4** and **S5**, Bull [14] and Fine [40] who investigated the logics containing **S4.3**, and Lemmon [86, 87, 88] and Segerberg [129] who launched a systematical investigation of various classes of modal logics. Two other influential figures that should also be mentioned here are Kuznetsov [81, 84, 85] and Jankov [67, 66, 68, 69] who investigated the class of all extensions of intuitionistic propositional logic which is closely related to the class of modal logics containing **S4**; see Section 9.

Although not formulated explicitly, the 'globalist's' dream research programme was to develop a mathematical machinery that could provide general solutions to the following major problems:[1]

1. given a class of models/structures, axiomatise the modal logic it determines, decide in an effective way whether it has certain important properties, say, decidability, compactness, interpolation, etc., and determine its computational complexity.

2. given a modal logic in the form of a finite set of axioms and inference rules, characterise the (simplest, smallest, largest, etc.) class of models/structures with respect to which this logic is sound and complete, decide in an effective way whether it has important properties as above, and determine its computational complexity.

This research programme is formulated in quite general terms and therefore can be interpreted in various ways. For example, it is not specified what kind of classes of frames/models we consider and what kind of axiomatic systems we take into account. Of course, different interpretations may lead to different solutions, but anyway first results within this ambitious programme looked very promising indeed! For example, Bull [14] proved that all extensions of **S4.3** have the finite model property and Fine [40] showed that all of them are finitely axiomatisable, and so decidable. (Actually, Dummett and Lemmon [33] claimed that all logics between **S4** and **S5** have the finite model property, but their proof was wrong: ten years later Jankov [68] constructed a counterexample.) In view of Makinson's theorem [94], one can effectively decide whether a given logic above **K** is consistent. Maksimova [95, 97] proved that two properties of logics containing **S4**—tabularity and interpolation—are decidable as well. It seems that many modal logicians did believe in an eventual success of this Big Programme.

In this chapter we analyse the development of Modal Logic within the research framework formulated above, starting from the beginning of the 1970s, although not necessarily in chronological order; for a historical analysis of mathematical modal logic the reader is referred to the recent paper of Goldblatt [57] and notes in [24]. Because of space limitations, we mainly concentrate on normal (multi-) modal logics and their decidability and completeness (in particular, with respect to Kripke or finite frames).

[1]Kuznetsov did formulate such problems *explicitly* in the context of superintuitionistic logics; e.g., given an axiomatisation of a superintuitionistic logic, can we decide in an effective way whether the logic is characterised by a finite algebra?

Roughly, our plan is as follows. We start in Section 2 with Thomason's explication (i′) of the *semantical* part (i) of the research programme above. Then, in Section 3, we lay the foundation for the most important *syntactical* notion of Modal Logic, namely, that of a *normal modal logic*. Having introduced an adequate semantics for normal modal logics in terms of *general frames*, we discuss in detail Blok's dichotomy in order to clarify the difference between Thomason's semantical definition of modal logics and the syntactically defined normal modal logics. Based on this discussion, we then come to the appropriate refinement (ii′) of the *syntactical* part (ii) of the research programme for normal modal logics and solutions to it given by Chagrov and Thomason.

Although beautiful from a mathematical point of view, the results of Thomason and Chagrov are 'negative' in the sense that almost all general algorithmic problems formulated in the Big Research Programme turn out to be undecidable. In the same way as the negative solution to the *classical decision problem* of Hilbert transformed the original problem into a *classification* problem, the 'negative' solution to the modal decision problems brings us down to a more 'modest' and realistic 'relativisation' of the programme to various syntactically or semantically defined classes of modal logics.

In Section 4, we consider logics axiomatised by formulas satisfying certain syntactical constraints, in particular, Sahlqvist formulas, uniform formulas, modal reduction principles, etc., and see whether such constraints allow us to prove general decidability/completeness results. In Section 5, we survey the literature on general decidability/completeness results for logics with some 'strong' axioms, say, extensions of tabular and pretabular logics, logics of finite depth and width, extensions of **S4.3**, **K5**, etc.

Then, in Section 6, we discuss an attempt to attack the Big Research Programme for normal extensions of **K4** (that is, unimodal logics with transitive general frames) and the tense logic **Lin** (of linear flows of time) by means of finite representations of modally definable classes of frames via frame and subframe formulas of Jankov and Fine [69, 41, 45] and more general 'canonical' formulas of [172, 174, 163]. This technique will be also used to draw and discuss connections between extensions of **S4** and superintuitionistic logics in Section 9.

In Section 7 we provide a 'positive' solution to the Big Research Programme for the class of all tense logics of linear flows of time. In fact, it turns out that for this class of logics all the questions posed in the programme are decidable (sometimes even in nondeterministic polynomial time).

In Section 8, we consider the class of subframe logics—i.e., logics determined by classes of (general) frames closed under the formation of substructures in the standard model-theoretic sense—and explore to what extent the research programme can be realised for this semantically defined class of modal logics.

A number of important open problems are formulated throughout the chapter.

2 THOMASON'S ANALYSIS

As we saw in Chapter 1, the standard propositional modal language with a countably infinite set of propositional variables (say, $p_0, p_1, \dots$), the Boolean connectives $\wedge$, $\neg$ (and their derivatives $\rightarrow$, $\vee$, etc.) and unary modal operators $\Box_1, \dots, \Box_n$ can be regarded as a basic tool for talking about relational structures $\mathfrak{F} = \langle W, R_1, \dots, R_n \rangle$, where the R_i are binary relations on $W \neq \emptyset$. We denote this *n-modal language* by $\mathcal{ML}_n$ and call $\mathfrak{F}$ an *n-frame* or simply a (*Kripke*) *frame*, if n is understood.

$\mathcal{ML}_n$ is interpreted in n-frames by means of *valuations* $\mathfrak{V}$ which associate with every propositional variable p_i a subset $\mathfrak{V}(p_i)$ of W. The pair $\mathfrak{M} = (\mathfrak{F}, \mathfrak{V})$ is called a (*Kripke*) *model* based on $\mathfrak{F}$. Given an $\mathcal{ML}_n$-formula φ, the *truth-relation* $(\mathfrak{M}, x) \models \varphi$, read as '*$\varphi$ is true at x in $\mathfrak{M}$*' (for $x \in W$), is defined by induction on the construction of φ as follows:

$$
\begin{array}{lll}
(\mathfrak{M}, x) \models p_i & \text{iff} & x \in \mathfrak{V}(p_i), \\
(\mathfrak{M}, x) \models \psi \wedge \chi & \text{iff} & (\mathfrak{M}, x) \models \psi \text{ and } (\mathfrak{M}, x) \models \chi, \\
(\mathfrak{M}, x) \models \neg\psi & \text{iff} & \text{not } (\mathfrak{M}, x) \models \psi, \\
(\mathfrak{M}, x) \models \Box_j\psi & \text{iff} & (\mathfrak{M}, y) \models \psi \text{ for all } y \in W \text{ such that } xR_jy.
\end{array}
$$

If $(\mathfrak{M}, x) \models \varphi$ does not hold then we write $(\mathfrak{M}, x) \not\models \varphi$ and say that $\mathfrak{M}$ *refutes* φ at x. Instead of $(\mathfrak{M}, x) \models \varphi$ and $(\mathfrak{M}, x) \not\models \varphi$ we write simply $x \models \varphi$ and $x \not\models \varphi$, if $\mathfrak{M}$ is understood. A formula φ is said to be *true in* $\mathfrak{M}$ ($\mathfrak{M} \models \varphi$, in symbols) if $x \models \varphi$ for all $x \in W$; φ is *satisfied in* $\mathfrak{M}$ if $x \models \varphi$ for some $x \in W$. We say that φ is *valid* in the frame $\mathfrak{F}$ (or $\mathfrak{F}$ *validates* φ) and write $\mathfrak{F} \models \varphi$ if φ is true in all models based on $\mathfrak{F}$; φ is *satisfiable in* $\mathfrak{F}$ if it is satisfied in some model based on $\mathfrak{F}$. For a set Γ of $\mathcal{ML}_n$-formulas, we say that $\mathfrak{F}$ is a *frame for* Γ if all formulas from Γ are valid in $\mathfrak{F}$. In this case we write $\mathfrak{F} \models \Gamma$. A formula φ is Γ-*satisfiable* if it is satisfiable in a frame for Γ. Finally, we write $\Gamma \models\!\!\!\!\!\!\;\approx \varphi$ if φ is valid in every frame for Γ (that is, if φ is a *semantic consequence of* Γ *over Kripke frames*).

As in classical first-order logic, given a class $\mathcal{K}$ of structures for our language—that is, a class of n-frames—we define the *theory* $\mathsf{Th}\,\mathcal{K}$ of $\mathcal{K}$ by the equation

$$\mathsf{Th}\,\mathcal{K} \quad = \quad \{\varphi \in \mathcal{ML}_n \mid \varphi \text{ is valid in every frame from } \mathcal{K}\}.$$

For example, as we know from Chapters 1 and 2, $\mathbf{K}_n$ is the theory of the class of all n-frames and **S4** is the theory of the class of all partial orders (or all quasi-orders).

Conversely, given a set Γ of $\mathcal{ML}_n$-formulas, denote by $\mathsf{Fr}\,\Gamma$ the class of frames for Γ. For example, $\mathsf{Fr}\,\{\Box p \to \Box\Box p\}$ is the class of all transitive frames $\mathfrak{F} = \langle W, R\rangle$. This and other similar results were in fact obtained by Kripke in his seminal paper [79].

Being equipped with these notions and notations, let us take a closer look at the research programme from Section 1. Clearly, problem (i) depends on *how exactly* the class of models/frames we are interested in is presented. For example, it can be given as the class of structures satisfying certain first- or second-order sentences. This understanding of (i) would lead us to the branch of Modal Logic known as *correspondence theory* (which is partially considered in Chapters 1 and 5; see also [153]). In the pure modal perspective, it makes sense to describe frame classes by means of modal formulas, namely as $\mathsf{Fr}\,\{\varphi\}$, for $\varphi \in \mathcal{ML}_n$. Thus, we arrive to our first precise approximation of (i):

(i′) given an arbitrary $\mathcal{ML}_n$-formula φ, axiomatise $\mathsf{Th}\,\mathsf{Fr}\,\{\varphi\} = \{\psi \in \mathcal{ML}_n \mid \varphi \models\!\!\!\!\!\!\;\approx \psi\}$, decide in an effective way whether $\mathsf{Th}\,\mathsf{Fr}\,\{\varphi\}$ has certain important properties, say, consistency, decidability, interpolation, etc., and determine its computational complexity.

Note that this problem is not as trivial as it might look from first sight. Of course, for numerous formulas φ the theory $\mathsf{Th}\,\mathsf{Fr}\,\{\varphi\}$ was axiomatised and thoroughly investigated a long time ago. Well-known examples are, for instance, $\mathbf{T} = \mathsf{Th}\,\mathsf{Fr}\,\{\Box p \to p\}$, the theory of the reflexive frames, and $\mathbf{K4} = \mathsf{Th}\,\mathsf{Fr}\,\{\Box p \to \Box\Box p\}$, the theory of the transitive

frames; see Chapters 1 and 2. Our concern here, however, is not some particular formulas, but effective axiomatisation procedures which could work for *all* modal formulas. Such a procedure is available, for example, for first-order logic: by Gödel's completeness theorem, given a first-order sentence φ, we can axiomatise the theory of structures validating φ by adding φ as an extra axiom to any standard Hilbert-style first-order system. But do we have a Gödel-type completeness theorem for $\mathcal{ML}_n$ with respect to Kripke frames?

A comprehensive analysis of problem (i′) was launched by S. Thomason in his series of papers [140, 141, 143, 145, 144].

THEOREM 1 (Thomason). (a) *There is an $\mathcal{ML}_1$-formula φ such that the set*

$$\mathsf{Th}\,\mathsf{Fr}\,\{\varphi\} \quad = \quad \{\psi \in \mathcal{ML}_1 \mid \varphi \models \psi\}$$

is Π^1_1-complete (in particular, it is not recursively enumerable).

(b) *There is no algorithm which is capable of deciding, given an $\mathcal{ML}_2$-formula φ, whether the theory $\mathsf{Th}\,\mathsf{Fr}\,\{\varphi\}$ is consistent, or, equivalently, whether there exists a 2-frame validating φ.*

(c) *There is a set Γ of $\mathcal{ML}_1$-formulas and an $\mathcal{ML}_1$-formula φ such that $\Gamma \models \varphi$, but $\Delta \not\models \varphi$ for any finite $\Delta \subset \Gamma$.*

(d) *For every $n < \omega + \omega$, there is an $\mathcal{ML}_2$-formula γ_n such that every rooted frame validating γ_n is of cardinality $\beth_n$ and, moreover, all such frames are isomorphic.*[2] *(Here $\beth_0 = \aleph_0$, $\beth_{m+1} = 2^{\beth_m}$, and $\beth_\omega = \lim\{\beth_m \mid m < \omega\}$.)*

Theorem 1 (a) shows that the first part of research problem (i′) cannot be solved, while both (a) and (b) indicate that the second part is also hopeless. In fact, this theorem clearly shows that modal theories behave similarly to second-order logic: we do not have any of the classical properties of first-order logic such as compactness, recursive enumerability of valid formulas or the Löwenheim–Skolem theorem. Interestingly, rather simple $\mathcal{ML}_1$-formulas φ such that $\mathsf{Fr}\,\{\varphi\}$ is not first-order definable had already been constructed by Segerberg [129]. Perhaps, the best known example is the *Löb formula*

$$\boldsymbol{la} = \Box(\Box p \to p) \to \Box p$$

which is valid in a frame $\mathfrak{F} = \langle W, R\rangle$ iff $\mathfrak{F}$ is transitive and contains no infinite ascending chain $x_1 R x_2 R x_3 \dots$. The latter condition is not definable in first-order logic.

Thomason proved his results by constructing (rather complex and 'artificial') multi-modal formulas with the required properties. To obtain (a) and (c) he showed that multi-modal logics can be reduced to unimodal logics with similar properties. (For a discussion of this kind of reductions see Chapter 8.) Now (in 2005) we know more simple and natural bimodal formulas with the properties of φ in (a) of Theorem 1. Take, for example, the conjunction φ of the following formulas:

$$\Box_1(\Box_1 p \to p) \to \Box_1 p, \qquad \Box_2(\Box_2 p \to p) \to \Box_2 p, \tag{1}$$

$$\Box_1(\Box_1 p \to q) \vee \Box_1(\Box_1 q \to p), \qquad \Box_2(\Box_2 p \to q) \vee \Box_2(\Box_2 q \to p), \tag{2}$$

$$\Diamond_2\Diamond_1 p \leftrightarrow \Diamond_1\Diamond_2 p, \qquad \Diamond_1\Box_2 p \to \Box_2\Diamond_1 p. \tag{3}$$

(1) says that both boxes satisfy the Löb axiom above, (2) ensures that frames for $\Box_1$ and $\Box_2$ satisfy the connectedness condition

$$\forall x, y, z\ (xR_i y \wedge xR_i z \wedge y \neq z \to yR_i z \vee zR_i y),$$

[2] See also [16] and [75].

while (3) says that R_1 and R_2 commute and satisfy the Church–Rosser property, i.e.,

$$\forall x, z\,\big(\exists y\,(xR_2y \land yR_1z) \leftrightarrow \exists y\,(xR_1y \land yR_2z)\big),$$
$$\forall x\forall y\forall z\,\big(xR_2y \land xR_1z \to \exists u\,(yR_1u \land zR_2u)\big).$$

Typical frames validating φ are products of finite strict linear orders or infinite reverse well-founded linear orders like $\langle\{0,1,\dots,n\},<\rangle$ or $\langle\omega+1,>\rangle$, where

$$\begin{array}{lcl} \langle W_h, R_h\rangle \times \langle W_v, R_v\rangle & = & \langle W_h \times W_v, R_1, R_2\rangle, \\ (u,v)R_1(u',v') & \text{iff} & uR_hu' \text{ and } v = v', \\ (u,v)R_2(u',v') & \text{iff} & vR_vv' \text{ and } u = u'. \end{array}$$

(For more details about product frames and logics see Chapter 15). To show that validity in frames for φ is Π^1_1-hard, one can reduce the following Σ^1_1-complete *recurrent tiling problem* [62] to the satisfiability problem for $\mathcal{ML}_2$-formulas in frames for φ: 'given a finite set T of tile types (1×1-squares with colours along their edges) and a $t_0 \in T$, can T tile the $\mathbb{N}\times\mathbb{N}$-grid in such a way that colours on adjacent edges of adjacent tiles match and t_0 appears infinitely often in the first column?' For details the reader is referred to [49, 120].

Of course, the notion of validity in Kripke frames is of a second-order nature. Indeed, we can easily define a translation $\cdot^s$ of $\mathcal{ML}_n$ into monadic second-order logic with n extra binary relations $R_1,\dots,R_n$ by taking inductively

$$\begin{array}{lcl} p_i^s & = & P_i(x) \\ (\varphi\land\psi)^s & = & \varphi^s \land \psi^s \\ (\neg\varphi)^s & = & \neg\varphi^s \\ (\Box_i\varphi)^s & = & \forall y\ (xR_iy \to \varphi^s[y/x]), \end{array}$$

where y is a fresh variable. And then, for every frame $\mathfrak{F} = \langle W, R_1,\dots,R_n\rangle$ and every $\mathcal{ML}_n$-formula $\varphi(p_1,\dots,p_k)$, we have $\mathfrak{F}\models\varphi$ iff $\forall P_1\dots\forall P_k\forall x\ \varphi^s$ is true in $\mathfrak{F}$. This means that for every set $\Gamma\cup\{\varphi\}$ of $\mathcal{ML}_n$-formulas, $\Gamma \mathrel{|\!\!\!\asymp} \varphi$ iff φ^s is a logical consequence of $\{\gamma^s \mid \gamma\in\Gamma\}$ in second-order logic.

What is more surprising is that the full monadic second-order theory of a binary predicate can be reduced to propositional modal logic:

THEOREM 2 (Thomason). *There is an effective translation $\cdot^\sharp$ of the language $\mathcal{MSO}$ of monadic second-order logic with one extra binary relation into $\mathcal{ML}_1$, and there is an $\mathcal{ML}_1$-formula τ such that, for every set $\Xi\cup\{\zeta\}$ of $\mathcal{MSO}$-formulas,*

$$\zeta \text{ is a logical consequence of } \Xi \quad \text{iff} \quad \{\tau\}\cup\{\xi^\sharp \mid \xi\in\Xi\} \mathrel{|\!\!\!\asymp} \zeta^\sharp.$$

Thus, Thomason's [145] conclusion was that $\mathcal{ML}_n$ or even $\mathcal{ML}_1$ can be regarded as a rather strong fragment of second-order predicate logic and that the modal consequence relation $\mathrel{|\!\!\!\asymp}$ is as complex as it could be. In particular, there is an $\mathcal{ML}_1$-formula φ such that the set $\{\psi\in\mathcal{ML}_1 \mid \varphi \mathrel{|\!\!\!\asymp} \psi\}$ is not definable in number theory of any finite order.

3 NORMAL MODAL LOGICS

The analogy with second-order logic discussed above also indicated a way of 'regaining' nice properties of first-order logic (compactness, recursive enumerability and Löwenheim–Skolem). Recall (see, e.g., [34]) that following Henkin's idea of introducing a special

universe over which the predicates can be interpreted, one can obtain an essentially first-order semantics—known as *general structures*—for second-order logic. Actually, this appears to be Thomason's [141] motivation for introducing *general frames* (which he called *first-order semantics* for modal logic).[3] The idea is very simple: restrict the range of the valuation function $\mathfrak{V}$ in the definition of Kripke models to some subset of 2^W that is closed under the available operations. This leads us to the following definition.

A *general n-frame* is a structure of the form

$$\mathfrak{G} \quad = \quad \langle W, R_1, \ldots, R_n, P \rangle ,$$

where $\langle W, R_1, \ldots, R_n \rangle$ is an ordinary Kripke frame and P is a subset of 2^W containing W and closed under set-theoretic intersection and complementation as well as under the operations

$$\Box_i X \quad = \quad \{x \in W \mid \forall y\ (xR_i y \to y \in X)\},$$

for all $i = 1, \ldots, n$. As before, the language $\mathcal{ML}_n$ is interpreted in general frames by means of *valuations* $\mathfrak{V}$ which associate with every propositional variable p_i a set $\mathfrak{V}(p_i)$. The only difference is that now $\mathfrak{V}(p_i)$ *must belong to* P. The pair $\mathfrak{M} = \langle \mathfrak{G}, \mathfrak{V} \rangle$ is called then a *model* based on the general frame $\mathfrak{G}$. The remaining semantical notions are defined in precisely the same way as for Kripke models, e.g., $\mathsf{Th}\,\mathcal{K}$, a general frame for a logic, etc. To simplify notation, we denote general frames of the form $\mathfrak{F} = \left\langle W, R_1, \ldots, R_n, 2^W \right\rangle$ by $\mathfrak{F} = \langle W, R_1, \ldots, R_n \rangle$ (because the theories of these frames are the same).

The first radical difference between Kripke and general frames is that the set of theories of classes of general frames can be characterised *syntactically*. Say that a subset L of $\mathcal{ML}_n$ is a *normal n-modal logic* if it contains the tautologies of classical propositional logic, the formulas

$$\Box_i (p \to q) \to (\Box_i p \to \Box_i q)$$

for all $i = 1, \ldots, n$, and is closed under the rules of uniform substitution, modus ponens and *necessitation* $\varphi / \Box_i \varphi$. The smallest normal n-modal logic is known to be $\mathbf{K}_n$; it clearly coincides with the theory of the class GFr of all general n-frames.

Now, given a set Γ of $\mathcal{ML}_n$-formulas, denote by $\mathbf{K}_n \oplus \Gamma$ the minimal normal n-modal logic containing Γ, and by $\mathsf{GFr}\,\Gamma$ the class of general frames for Γ.

THEOREM 3. (a) *For every class $\mathcal{K}$ of general n-frames,* $\mathsf{Th}\,\mathcal{K}$ *is a normal n-modal logic. More precisely,*

$$\mathsf{Th}\,\mathcal{K} \quad = \quad \mathbf{K}_n \ \oplus\ \{\varphi \in \mathcal{ML}_n \mid \varphi \textit{ is valid in every general frame from } \mathcal{K}\}.$$

(b) *For every set Γ of $\mathcal{ML}_n$-formulas,*

$$\mathbf{K}_n \ \oplus\ \Gamma \quad = \quad \mathsf{Th}\,\mathsf{GFr}\,\Gamma.$$

(Some comments on the proof of this theorem will be given later on in this section; see also Chapter 5.)

From now on, instead of $\mathsf{Th}\,\mathcal{K}$ we will write $\mathsf{Log}\,\mathcal{K}$ and call it the (*normal n-modal*) *logic characterised* (or *determined*) *by* $\mathcal{K}$. The set of all normal n-modal logics containing

[3] In fact, general frames were also introduced in 1951 by Jónsson and Tarski [71] as Stone-like representations of Boolean algebras with operators; see [57] for more references, details and discussion.

a logic L will be denoted by $\mathrm{NExt}\, L$. In particular, $\mathrm{NExt}\, \mathbf{K}_n$ is the set of all normal n-modal logics. Thus,

$$\mathrm{NExt}\, \mathbf{K}_n \quad = \quad \{\mathbf{K}_n \oplus \Gamma \mid \Gamma \subseteq \mathcal{ML}_n\} \quad = \quad \{\mathsf{Log}\, \mathcal{K} \mid \mathcal{K} \subseteq \mathsf{GFr}\,\}.$$

Let Kripke_n be the set of all *Kripke complete* normal n-modal logics, that is

$$\mathsf{Kripke}_n \quad = \quad \{\mathsf{Log}\, \mathcal{K} \mid \mathcal{K} \text{ a class of Kripke } n\text{-frames}\}.$$

As follows from Theorem 1 (a), Kripke_n is a *proper* subset of $\mathrm{NExt}\, \mathbf{K}_n$. Indeed, take a formula φ such that $\mathsf{Log}\,\mathsf{Fr}\,\{\varphi\}$ is not recursively enumerable and consider the normal modal logic $\mathbf{K}_n \oplus \varphi$. Then

$$\mathsf{Log}\,\mathsf{Fr}\,\{\varphi\} \quad \supsetneqq \quad \mathsf{Log}\,\mathsf{GFr}\,\{\varphi\} \quad = \quad \mathbf{K}_n \ \oplus \ \varphi$$

simply because $\mathbf{K}_n \oplus \varphi$ is recursively enumerable. On the other hand, $\mathsf{Log}\,\mathsf{Fr}\,\{\varphi\}$ and $\mathsf{Log}\,\mathsf{GFr}\,\{\varphi\}$ have precisely the same Kripke frames, and so the latter cannot be Kripke complete. The logic $\mathbf{K}_2 \oplus (1) \oplus (2) \oplus (3)$ is probably the 'simplest natural' Kripke incomplete normal modal logic.

So what is the relation between the classes Kripke_n and $\mathrm{NExt}\, \mathbf{K}_n$? What can be said about the class $\mathrm{NExt}\, \mathbf{K}_n - \mathsf{Kripke}_n$?

3.1 Blok's dichotomy

The first examples of Kripke incomplete logics were discovered by Thomason [141, 142] and Fine [42]. In order to understand the phenomenon of Kripke incompleteness more deeply, Fine proposed to investigate how many logics may share the same Kripke frames with a given normal (uni)modal logic L. The cardinality of the set

$$\{L' \in \mathrm{NExt}\, \mathbf{K} \mid \mathsf{Fr}\, L \ = \ \mathsf{Fr}\, L'\}$$

was called by Fine the *degree of Kripke incompleteness* of L. A very interesting complete solution to this problem was found by Blok [8]. The key player in his solution was the concept of splitting originating in lattice theory [106] (for details see Chapter 8).

To explain the idea behind Blok's result informally, let us observe first that a Kripke complete logic L is always the maximal logic in the set $\{L' \mid \mathsf{Fr}\, L' = \mathsf{Fr}\, L\}$. Now suppose that L is a Kripke complete logic with the following property: there exists a Kripke frame $\mathfrak{F}$ such that L is the smallest logic in the class $\{L' \in \mathrm{NExt}\, \mathbf{K} \mid \mathfrak{F} \notin \mathsf{Fr}\, L'\}$. Then the degree of Kripke incompleteness of L is 1. Indeed, assume that L' is a normal modal logic with $\mathsf{Fr}\, L = \mathsf{Fr}\, L'$. Then $\mathfrak{F} \notin \mathsf{Fr}\, L'$, and so $L \subseteq L'$. To prove $L' \subseteq L$, assume $\varphi \notin L$. As L is Kripke complete, there exists a Kripke frame $\mathfrak{F}' \in \mathsf{Fr}\, L$ such that $\mathfrak{F}' \not\models \varphi$. And since $\mathfrak{F}' \in \mathsf{Fr}\, L'$, we then have $\varphi \notin L'$. Of course, the same argument goes through if instead of just one frame $\mathfrak{F}$ we take some set of frames.

Thus, we can try to generate modal logics with degree of Kripke incompleteness 1 by taking sets $\mathcal{F}$ of frames, proving that the smallest normal logic $L_{\mathcal{F}}$ in the class $\{L' \in \mathrm{NExt}\, \mathbf{K} \mid \exists \mathfrak{F} \in \mathcal{F}\, \mathfrak{F} \notin \mathsf{Fr}\, L'\}$ *exists* and showing its Kripke completeness. Blok's achievement was that he (a) characterised sets $\mathcal{F}$ of frames for which the logics $L_{\mathcal{F}}$ exist, (b) proved that all these $L_{\mathcal{F}}$ are Kripke complete (actually, have the finite model property), and (c) showed that any normal modal logic different from such $L_{\mathcal{F}}$ has degree of Kripke incompleteness $2^{\aleph_0}$.

We now introduce the notions required to explain Blok's result in more detail. Given a logic $L_0 \in \mathrm{NExt}\,\mathbf{K}$, we say that a (finite rooted) frame $\mathfrak{F}$ *splits* $\mathrm{NExt}\,L_0$ if $\mathfrak{F}$ is not a frame for the normal modal logic

$$L_1 \quad = \quad \bigcap\{L \in \mathrm{NExt}\,L_0 \mid \mathfrak{F} \not\models L\}.$$

In this case we denote L_1 by $L_0/\mathfrak{F}$ and call it the *splitting* of $\mathrm{NExt}\,L_0$ *by* $\mathfrak{F}$. This notation reflects the fact that $L_0/\mathfrak{F}$ is the *smallest* logic in $\mathrm{NExt}\,L_0$ which does not have $\mathfrak{F}$ as its frame. If all frames in a set $\mathcal{F}$ split $\mathrm{NExt}\,L_0$, we call $\bigoplus\{L_0/\mathfrak{F} \mid \mathfrak{F} \in \mathcal{F}\}$—i.e., the smallest normal modal logic containing $\bigcup\{L_0/\mathfrak{F} \mid \mathfrak{F} \in \mathcal{F}\}$—the *union-splitting of* $\mathrm{NExt}\,L_0$ *by* $\mathcal{F}$ and denote it by $L_0/\mathcal{F}$.

EXAMPLE 4. Denote by $\bullet$ the Kripke frame which consists of a single irreflexive point. A frame comprised of a single reflexive point is denoted by $\circ$.

(a) $\bullet$ splits $\mathrm{NExt}\,\mathbf{K}$ and $\mathbf{D} = \mathbf{K}/\bullet$ (we remind the reader that $\mathbf{D} = \mathbf{K} \oplus \Diamond\top$ is characterised by the class of *serial* Kripke frames in which every point has a successor). To see this, set

$$L_1 \quad = \quad \bigcap\{L \in \mathrm{NExt}\,\mathbf{K} \mid \bullet \not\models L\}.$$

Since, for every $L \in \mathrm{NExt}\,\mathbf{K}$, $\bullet \not\models L$ iff $\Diamond\top \in L$, L_1 is the intersection of all normal modal logics containing $\Diamond\top$. But $\mathbf{D}$ is the smallest such logic and therefore $L_1 = \mathbf{D}$.

(b) $\circ$ does not split $\mathrm{NExt}\,\mathbf{K}$. To see this recall that $\mathbf{K}$ is determined by the class of finite frames $\mathfrak{F} = \langle W, R\rangle$ without cycles (i.e., R-paths from a point to itself); see, e.g., Chapter 1. For every such $\mathfrak{F}$, we have $\circ \not\models \mathsf{Log}\,\{\mathfrak{F}\}$ because there exists $n < \omega$ such that $\Box^n\bot \in \mathsf{Log}\,\{\mathfrak{F}\}$, but $\circ \not\models \Box^n\bot$. Therefore,

$$\bigcap\{L \in \mathrm{NExt}\,\mathbf{K} \mid \circ \not\models L\} \quad \subseteq \quad \bigcap\{\mathsf{Th}\,\mathfrak{F} \in \mathrm{NExt}\,\mathbf{K} \mid \mathfrak{F} \text{ finite and cycle free }\} \quad = \quad \mathbf{K}$$

which means that there does *not* exist a smallest normal modal logic without $\circ$ among its frames.

(c) No frame with cycles splits $\mathrm{NExt}\,\mathbf{K}$. The argument is similar to that in (b): just use the fact that no $\Box^n\bot$ is valid in a frame with cycles.

(d) Every finite cycle-free rooted frame splits $\mathrm{NExt}\,\mathbf{K}$. To prove this, we associate with every finite rooted frame $\mathfrak{F} = \langle W, R\rangle$ the formula

$$\delta_{\mathfrak{F}} \quad = \quad \bigwedge_{xRy}(p_x \to \Diamond p_y) \wedge \bigwedge_{\neg xRy}(p_x \to \neg\Diamond p_y) \wedge \bigwedge_{x \neq y}(p_x \to \neg p_y) \wedge \bigvee_{x \in W} p_x.$$

Suppose now that $\mathfrak{F}$ is cycle free, r is a root of $\mathfrak{F}$, $d(\mathfrak{F})$ is the *depth* of $\mathfrak{F}$ (i.e., the length of the longest R-path in $\mathfrak{F}$), and $\Box^{\leq n}\varphi = \varphi \wedge \Box\varphi \wedge \cdots \wedge \Box^n\varphi$. It is not hard to see then that a (general) frame $\mathfrak{G}$ satisfies $\Box^{\leq d(\mathfrak{F})}\delta_{\mathfrak{F}} \wedge p_r$ iff there is a generated subframe $\mathfrak{H}$ of $\mathfrak{G}$ which can be p-morphically mapped onto $\mathfrak{F}$. It follows that the smallest normal logic without $\mathfrak{F}$ among its frames exists and can be axiomatised as

$$\mathbf{K}/\mathfrak{F} \quad = \quad \mathbf{K} \oplus \Box^{\leq d(\mathfrak{F})}\delta_{\mathfrak{F}} \to \neg p_r. \tag{4}$$

(e) The *inconsistent logic*—i.e., $\mathcal{ML}_1$—can be represented as $\mathbf{D}/\circ = (\mathbf{K}/\bullet)/\circ$ (actually this is a variant of Makinson's theorem [94]).

(f) Note by the way that if $L_0 \in \mathrm{NExt}\,\mathbf{K4}$ then *every* finite rooted transitive frame $\mathfrak{F}$ for L_0 splits $\mathrm{NExt}\,L_0$ and $L_0/\mathfrak{F} = L_0 \oplus \Box^{\leq 1}\delta_{\mathfrak{F}} \to \neg p_r$ (a (general) transitive frame $\mathfrak{G}$

satisfies $\Box^{\leq 1}\delta_{\mathfrak{F}} \wedge p_r$ iff there is a generated subframe $\mathfrak{H}$ of $\mathfrak{G}$ which can be p-morphically mapped onto $\mathfrak{F}$).

Now, returning back to the degree of Kripke incompleteness, we obtain the first part of Blok's dichotomy:

THEOREM 5 (Blok). (i) *A finite rooted frame $\mathfrak{F}$ splits* NExt **K** *iff it is cycle free. In this case we have $\mathbf{K}/\mathfrak{F} = \mathbf{K} \oplus \Box^{\leq n}\delta_{\mathfrak{F}} \to \neg p_r$, where $n = d(\mathfrak{F})$.*

(ii) *Every union-splitting of* NExt **K** *has the finite model property, and so its degree of Kripke incompleteness is* 1.

The proof of (ii) is by a variant of standard filtration (see Chapter 1). By (e) from the example above, the inconsistent logic $\mathcal{ML}_1$ also has degree of Kripke incompleteness 1 (it may be of interest to note that the degree of Kripke incompleteness of $\mathcal{ML}_2$ in NExt $\mathbf{K}_2$ is $2^{\aleph_0}$). The second part of Blok's dichotomy states that all normal modal logics not covered by Theorem 5 have degree of Kripke incompleteness $2^{\aleph_0}$:

THEOREM 6 (Blok). *If a logic L is inconsistent or a union-splitting of* NExt **K**, *then L has degree of Kripke incompleteness* 1. *Otherwise L has degree of Kripke incompleteness* $2^{\aleph_0}$ *in* NExt **K**.

Before we sketch a proof of this result it is worth spending some time on its interpretation. First, it means that **D** is the only 'standard' normal modal logic with degree of Kripke incompleteness 1. Logics like **S5**, **T**, **K4**, and **S4** have degree of incompleteness $2^{\aleph_0}$. In fact, every consistent normal logic containing **K4** or containing **D** properly as well as every consistent tabular normal modal logic (a logic is *tabular* if it is determined by a single finite frame; see Section 5), has degree of Kripke incompleteness $2^{\aleph_0}$. Second, in frame-theoretic terms it means that for every modally definable class $\mathcal{F}$ of, say, transitive frames (that is, $\mathcal{F} = \mathsf{Fr}\,\Gamma$ for some set Γ of modal formulas containing $\Box p \to \Box\Box p$) there exist uncountably many different $L \in$ NExt **K** such that

$$\mathcal{F} \quad = \quad \mathsf{Fr}\,L.$$

This applies, for example, to the class of all frames based on equivalence relations, quasi-orders, linear orders, and so on.

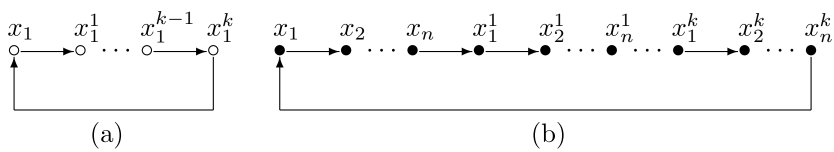


Figure 1.

Proof. Suppose that a consistent L is not a union-splitting and L' is the greatest union-splitting contained in L. Since L' has the finite model property, there is a finite rooted frame $\mathfrak{F} = \langle W, R\rangle$ for L' refuting some $\varphi \in L$ and such that every proper generated subframe of $\mathfrak{F}$ validates L. Clearly, $\mathfrak{F}$ is not cycle free. Let $x_1 R x_2 R \ldots R x_n R x_1$ be the shortest cycle in $\mathfrak{F}$ and $k = md(\varphi)+1$. We construct a new frame $\mathfrak{F}'$ by extending the cycle $x_1, \ldots, x_n, x_1$ as shown in Fig. 1 ((a) for $n = 1$ and (b) for $n > 1$). More precisely, we add to $\mathfrak{F}$ copies $x_i^1, \ldots, x_i^k$ of x_i for each $i \in \{1, \ldots, n\}$, organise them into the nontransitive cycle shown in Fig. 1 and draw an arrow from x_i^j to $y \in W - \{x_1, \ldots, x_n\}$ iff $x_i R y$.

Denote the resulting frame by $\mathfrak{F}' = \langle W', R' \rangle$ and let $x' = x_n^k$. By the construction, $\mathfrak{F}$ is a p-morphic image of $\mathfrak{F}'$. Therefore, for all models $\mathfrak{M} = \langle \mathfrak{F}, \mathfrak{V} \rangle$ and $\mathfrak{M}' = \langle \mathfrak{F}', \mathfrak{V}' \rangle$ such that

$$\mathfrak{V}'(p) \quad = \quad \mathfrak{V}(p) \cup \{x_i^j \mid x_i \in \mathfrak{V}(p),\ j < k\}$$

and for every $x \in W$ and every subformula ψ of φ, we have $(\mathfrak{M}, x) \models \psi$ iff $(\mathfrak{M}', x) \models \psi$. So we can hook some other model on x', and points in W will not feel its presence by means of φ's subformulas. The frame to be hooked on x' depends on whether $\bullet \models L$ or $\circ \models L$. We consider only the former alternative.

Fix some $m > |W'|$. For each $I \subseteq \omega - \{0\}$, let $\mathfrak{F}_I = \langle W_I, R_I, P_I \rangle$ be the frame whose diagram is shown in Fig. 2 (d_0 sees the root of $\mathfrak{F}'$, all points e_i and e'_j and is seen from x'; the subframes in dashed boxes are transitive, $e'_i \in W_I$ iff $i \in I$, and P_I consists of sets of the form $X \cup Y$ such that X is a finite or cofinite subset of $W_I - \{b, a_i \mid i < \omega\}$ and Y is either a finite subset of $\{a_i \mid i < \omega\}$ or is of the form $\{b\} \cup Y'$, where Y' is a cofinite subset of $\{a_i \mid i < \omega\}$. It is not hard to see that the points a_i, c, e_i and e'_i are

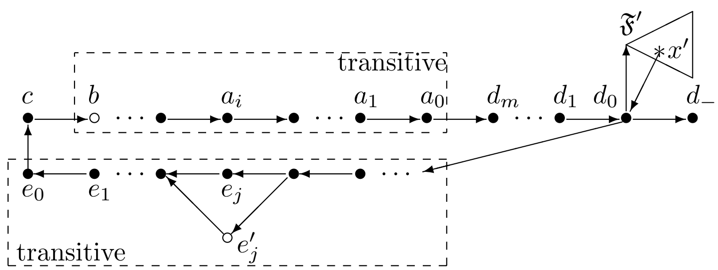


Figure 2.

characterised by the variable free formulas

$$\alpha_0 \;=\; \Diamond(\delta_m \wedge \Diamond(\delta_{m-1} \wedge \cdots \wedge \Diamond\delta_0)\cdots) \wedge \neg\Diamond^2(\delta_m \wedge \Diamond(\delta_{m-1} \wedge \cdots \wedge \Diamond\delta_0)\cdots),$$

$$\alpha_{i+1} \;=\; \Diamond\alpha_i \wedge \neg\Diamond^2\alpha_i, \qquad \gamma \;=\; \Diamond^2\alpha_0 \wedge \neg\Diamond\alpha_0,$$

$$\epsilon_0 \;=\; \Diamond\gamma, \qquad \epsilon_{i+1} \;=\; \Diamond\epsilon_i \wedge \neg\Diamond^2\epsilon_i, \qquad \epsilon'_{i+1} \;=\; \Diamond\epsilon_i \wedge \neg\Diamond^+\epsilon_{i+1},$$

(in the sense that $x \models \alpha_i$ iff $x = a_i$, etc.), where

$$\delta_0 \;=\; \Diamond\Box\bot, \qquad \delta_1 \;=\; \Diamond\delta_0 \wedge \neg\delta_0, \qquad \delta_2 \;=\; \Diamond\delta_1 \wedge \neg\delta_1 \wedge \neg\Diamond^+\delta_0,$$

$$\delta_{k+1} \;=\; \Diamond\delta_k \wedge \neg\delta_k \wedge \neg\Diamond^+\delta_{k-1} \wedge \cdots \wedge \neg\Diamond^+\delta_0.$$

Define L_I to be the logic determined by the class of frames for L and $\mathfrak{F}_I$, that is, $L_I = L \cap \mathsf{Log}\,\mathfrak{F}_I$. Since $\neg(\epsilon'_i \wedge \Diamond^{m+6}\neg\varphi) \in L_J - L_I$ for $i \in I - J$ (φ is refuted at the root of $\mathfrak{F}'$), $\left|\{L_I \mid I \subseteq \omega - \{0\}\}\right| = 2^{\aleph_0}$.

Let us show now that L_I has the same Kripke frames as L. Since $L_I \subseteq L$, we must prove that every Kripke frame for L_I validates L. Suppose there is a rooted Kripke frame $\mathfrak{G}$ such that $\mathfrak{G} \models L_I$ but $\mathfrak{G} \not\models \psi$, for some $\psi \in L$. Since ψ is in L, it is valid in all frames for L, in particular, $\bullet \models \psi$. And since $\psi \notin L_I$, ψ is refuted in $\mathfrak{F}_I$. Moreover, by the construction of $\mathfrak{F}_I$, it is refuted at a point from which the root of $\mathfrak{F}'$ can be reached by

a finite number of steps. Therefore, the following formulas are valid in $\mathfrak{F}_I$ and so belong to L_I and are valid in $\mathfrak{G}$:

$$\neg\psi \to \bigvee_{i=0}^{l} \Diamond^i \gamma, \tag{5}$$

$$\neg\psi \to \bigwedge_{i=0}^{l} \Box^i \big(\gamma \to \Box(\Box_0(\Box_0 p \to p) \to p)\big), \tag{6}$$

where p does not occur in ψ and l is a sufficiently big number so that any point in $\mathfrak{F}_I$ is accessible by $\leq l$ steps from every point in the selected cycle and every point at which ψ may be false, and $\Box_0\chi = \Box(\Diamond\alpha_0 \to \chi)$. According to (5), $\mathfrak{G}$ contains a point where γ is true. By the construction of γ, this point has a successor y where, by (6), $\Box_0(\Box_0 p \to p) \to p$ is true *under any valuation* in $\mathfrak{G}$ and $y \models \Diamond\alpha_0$. Define a valuation $\mathfrak{U}$ in $\mathfrak{G}$ by taking $\mathfrak{U}(p) = y\!\uparrow$, where $y\!\uparrow$ is the set of all points accessible from y. Then $y \models \Box_0(\Box_0 p \to p)$, from which $y \models p$ and so $y \in y\!\uparrow$. Now define another valuation $\mathfrak{U}'$ so that $\mathfrak{U}'(p) = y\!\uparrow - \{y\}$. Since y is reflexive, we again have $y \models \Box_0(\Box_0 p \to p)$, whence $y \models p$, which is a contradiction. ❑

Blok's dichotomy can be generalised in various directions. First, it holds for the languages $\mathcal{ML}_n$ and the corresponding classes NExt $\mathbf{K}_n$ as well; see [92]. And second, it can be extended to completeness with respect to the neighbourhood semantics [27] as well as some other, algebraically motivated, semantics for normal modal logics [92]. On the other hand, the following major problem remains open:

PROBLEM 1. Characterise the degree of Kripke incompleteness of 'transitive' logics in the classes NExt $\mathbf{K4}$, NExt $\mathbf{S4}$, etc., where Theorem 5 does not hold.

One conclusion to be drawn from these results is that Kripke complete logics are rather exceptional, that Kripke completeness of syntactically defined 'standard' modal logics is a kind of good luck. Another conclusion is that instead of considering logics in the class Kripke_n it may be worthwhile to move to the larger class NExt $\mathbf{K}_n$. First, as we know from other disciplines, more general settings can be very useful (for example, various problems about natural or rational numbers can only be analysed in the framework of real numbers). The second reason is that NExt $\mathbf{K}_n$ is quite natural not only from the syntactical point of view. In fact, as follows from Chapters 6 and 8, the lattice $\langle \text{NExt}\,\mathbf{K}_n, \subseteq\rangle$ is dually isomorphic to the lattice of varieties (alias equational theories) of Boolean algebras with operators. This means, in particular, that ideas and techniques from universal algebra are more suitable for investigating NExt $\mathbf{K}_n$ rather than Kripke_n.

Thus, it makes sense to extend the research programme above to the class of all normal n-modal logics.

Research programme for normal modal logics

Within the framework of normal modal logics, the original research programme can be interpreted as follows. By Thomason's Theorem 1, we know that (i′) is not realisable. The reformulation of (i′) for general frames has a trivial solution—just remember that $\mathsf{Log}\,\mathsf{GFr}\,\{\varphi\}$ is axiomatised as $\mathbf{K}_n \oplus \varphi$. Instead, we suggest the following reformulation a solution to which would clearly show how complex it is to axiomatise logics determined by classes of (general) frames:

(i″) Characterise those modal formulas φ for which we can effectively recognise whether $\mathbf{K} \oplus \psi$ axiomatises $\mathsf{Log\,GFr}\,\{\varphi\}$. Characterise those formulas φ for which we can effectively recognise whether $\mathbf{K} \oplus \psi$ axiomatises $\mathsf{Log\,Fr}\,\{\varphi\}$. For example, is there an algorithm which decides, for a formula ψ, whether $\mathbf{K} \oplus \psi$ axiomatises the logic of all transitive frames (i.e., $\mathsf{Log\,Fr}\,\{\Box p \to \Box\Box p\}$), reflexive frames (i.e., $\mathsf{Log\,Fr}\,\{\Box p \to p\}$), etc.?

The first part of (i″) can be reformulated as an axiomatisation problem. Given a modal formula φ and a logic $L_0 \in \mathrm{NExt}\,\mathbf{K}$, we say that the *axiomatisation problem* for $L_0 \oplus \varphi$ is *decidable above* L_0 if the set $\{\psi \in \mathcal{ML}_1 \mid L_0 \oplus \psi = L_0 \oplus \varphi\}$ is recursive. Then the first part of (i″) asks for a characterisation of those modal formulas φ for which the axiomatisation problem for $\mathbf{K} \oplus \varphi$ is decidable.

Being equipped with the notion of a normal modal logic, we can give a precise interpretation of the syntactically formulated problem (ii) from Section 1:

(ii″) Given a modal formula φ, characterise the (simplest, smallest, largest, etc.) class of frames with respect to which $\mathbf{K} \oplus \varphi$ is sound and complete. In particular, is it decidable whether the logic $\mathbf{K}\oplus\varphi$ is Kripke complete, has the finite model property, is determined by a finite frame? Furthermore, can we effectively recognise, given a modal formula φ, whether $\mathbf{K} \oplus \varphi$ is decidable, compact, has interpolation, etc.?

3.2 Chagrov's classification

Rather surprisingly, the *partition* of NExt $\mathbf{K}$ into union-splittings and non-union-splittings not only gives a concise solution to the problem of locating Kripke_n within NExt $\mathbf{K}_n$, but also provides means to attack (i″). A comprehensive solution was found by A. Chagrov.

To explain the intuition behind Chagrov's classification result for the axiomatisation problem, consider the logic $\mathbf{D} = \mathbf{K}/\bullet$ and suppose that we want to decide, for a given formula ψ, whether $\mathbf{K}/\bullet = \mathbf{K} \oplus \psi$ or, equivalently, whether

(a) $\mathbf{K} \oplus \psi \quad \subseteq \quad \mathbf{K}/\bullet$ and

(b) $\mathbf{K}/\bullet \quad \subseteq \quad \mathbf{K} \oplus \psi$.

Now, (a) is equivalent to the problem '$\psi \in \mathbf{K}/\bullet$?' which is decidable because the modal logic $\mathbf{D}$ is decidable. And (b) can be checked effectively because, by the definition of splittings, it is equivalent to the problem '$\bullet \not\models \psi$?'. Thus we have proved the decidability of the axiomatisation problem for $\mathbf{D}$ using the fact that $\mathbf{D}$ is decidable and is a union-splitting of NExt $\mathbf{K}$. By Theorem 5, all union-splittings have the finite model property and, therefore, are decidable if finitely axiomatisable. So this proof shows the decidability of the axiomatisation problem for every union-splitting $\mathbf{K}/\mathcal{F}$, where $\mathcal{F}$ is a finite set of finite rooted cycle free frames. Chagrov's achievement was to show that for no other logic is the axiomatisation problem decidable:

THEOREM 7 (Chagrov). *The axiomatisation problem for a consistent logic* $\mathbf{K} \oplus \varphi \supsetneq \mathbf{K}$ *is decidable iff* $\mathbf{K} \oplus \varphi$ *is a union-splitting.*

Hence, the axiomatisation problem is undecidable for such intuitively simple logics as $\mathbf{S4}$ or $\mathbf{K4}$. Again, $\mathbf{D}$ is the only 'standard' modal logic for which the problem is decidable. Before explaining the proof in some detail, we draw conclusions regarding the second part of (i″) and formulate and discuss solutions for (ii″).

COROLLARY 8. *If* $\mathsf{Fr}\{\varphi\}$ *is nonempty, then the problem '*$\mathbf{K} \oplus \psi = \mathsf{Log}\,\mathsf{Fr}\{\varphi\}$*?' is decidable iff*

- $\mathsf{Log}\,\mathsf{Fr}\{\varphi\}$ *is a union-splitting of* $\mathrm{NExt}\,\mathbf{K}$ *or*
- $\mathsf{Log}\,\mathsf{Fr}\{\varphi\}$ *is not finitely axiomatisable.*

Similarly to Blok's dichotomy, this means, for example, that for any nonempty modally definable class $\mathcal{F}$ of transitive frames such that $\mathsf{Log}\,\mathcal{F}$ is finitely axiomatisable, no algorithm can recognise whether a given formula axiomatises $\mathsf{Log}\,\mathcal{F}$.

The following result due to Thomason [146] and Chagrov [18, 19, 22] (for more details see [24] and references therein) gives 'negative' solutions to (ii$''$):

THEOREM 9 (Thomason & Chagrov). *The following sets are undecidable:*

(a) $\{\varphi \in \mathcal{ML}_1 \mid \mathbf{K} \oplus \varphi$ *is Kripke complete*$\}$,

(b) $\{\varphi \in \mathcal{ML}_1 \mid \mathbf{K} \oplus \varphi$ *is decidable*$\}$,

(c) $\{\varphi \in \mathcal{ML}_1 \mid \mathbf{K} \oplus \varphi$ *has the fmp*$\}$,

(d) $\{\varphi \in \mathcal{ML}_1 \mid \mathbf{K} \oplus \varphi$ *is tabular*$\}$,

(e) $\{\varphi \in \mathcal{ML}_1 \mid \mathbf{K} \oplus \varphi = L\}$, *where L is an arbitrarily fixed consistent tabular logic.*

Proof. We begin by showing how to prove Theorem 7. The implication ($\Leftarrow$) should be clear from the example $\mathbf{D} = \mathbf{K}/\bullet$ discussed above.

($\Rightarrow$) We show that if $L = \mathbf{K} \oplus \varphi \neq \mathbf{K}$ is a consistent logic that is not a union-splitting then the axiomatisation problem for L is undecidable. The proof is by reduction of the undecidable configuration problem for Minsky (alias register) machines with two tapes (registers).

We remind the reader that a *Minsky machine* (with two tapes) is a finite set of instructions for transforming triples $\langle s, m, n\rangle$ of natural numbers, called *configurations.* The intended meaning of the current configuration $\langle s, m, n\rangle$ is as follows: s is the number (label) of the current machine state and m, n represent the current state of information. Each instruction has one of the four possible forms:

$$s \to \langle t, 1, 0\rangle\,, \quad s \to \langle t, 0, 1\rangle\,, \quad s \to \langle t, -1, 0\rangle\,(\langle t', 0, 0\rangle), \quad s \to \langle t, 0, -1\rangle\,(\langle t', 0, 0\rangle).$$

The last of them, for instance, means: transform $\langle s, m, n\rangle$ into $\langle t, m, n-1\rangle$ if $n > 0$ and into $\langle t', m, n\rangle$ if $n = 0$. For a Minsky machine $\boldsymbol{P}$, we write $\boldsymbol{P} : \langle s, m, n\rangle \to \langle t, k, l\rangle$ if starting with $\langle s, m, n\rangle$ and applying the instructions in $\boldsymbol{P}$, in finitely many steps (possibly, in 0 steps) we can reach $\langle t, k, l\rangle$.

We use the well known fact (see, e.g., [102]) that the following *configuration problem* is undecidable: given a program $\boldsymbol{P}$ and configurations $\langle s, m, n\rangle$, $\langle t, k, l\rangle$, determine whether $\boldsymbol{P} : \langle s, m, n\rangle \to \langle t, k, l\rangle$.

Now let $L = \mathbf{K} \oplus \varphi$. Similarly to the proof of Blok's Theorem 6, we analyse two cases: $\bullet \models \varphi$ and $\circ \models \varphi$. Here we only show that the axiomatisability problem for $L = \mathbf{K} \oplus \varphi$ with $\bullet \models \varphi$ is undecidable and leave the remaining case to the reader.

We will use a modification of the (general) frame $\mathfrak{F}_I$ constructed in the proof of Blok's theorem. Let us take another look at this frame. The root of $\mathfrak{F}'$ refutes some formula

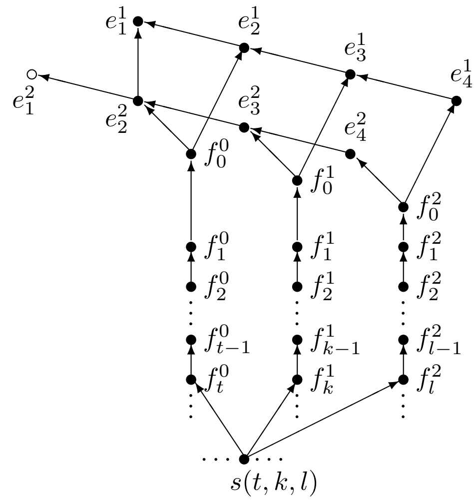


Figure 3.

in L; without loss of generality we may assume that this formula is φ. The part of $\mathfrak{F}_I$ comprising points a_i, b, c was used to ensure that the logic of $\mathfrak{F}_I$ is Kripke incomplete, while the points e_0, e_j, e'_j $(j > 0)$ were used to produce a continuum of frames of the form $\mathfrak{F}_I$, i.e., to make sure that $\mathsf{Log}\,\{\mathfrak{F}_I\} \neq \mathsf{Log}\,\{\mathfrak{F}_J\}$ whenever $I \neq J$. Finally, the points d_k $(-1 \leq k \leq m)$ ensured, in particular, that a_i, b, c, e_0, e_j, e'_j have no impact on $\mathfrak{F}'$ and the other way round.

For our present aims the subframe consisting of points a_i, b, c, e_0, e_j, e'_j is not required and we can modify it. Namely, we remove from $\mathfrak{F}_I$ the points e_j, e'_j $(j \geq 1)$ and replace them with the frame shown in Fig. 3. More precisely, the frame in Fig. 3 is transitive, e_1^2 is its only reflexive point, all points are accessible from d_0 and 'see' e_0; the points of the form $s(t,k,l)$ to not see each other and occur in the frame iff $\boldsymbol{P} : \langle s, m, n\rangle \to \langle t, k, l\rangle$, where $\boldsymbol{P}$, $\langle s, m, n\rangle$ are some fixed Minsky machine and configuration. Denote the resulting frame by $\mathfrak{F}(\boldsymbol{P}, \langle s, m, n\rangle)$.

Now, starting with the formula ϵ_0 from the proof of Blok's theorem, we define the following formulas

$$
\begin{array}{rclcrcl}
\epsilon_1^1 & = & \Diamond\epsilon_0 \wedge \neg\Diamond\Diamond\epsilon_0, & & \epsilon_1^2 & = & \Diamond\Diamond\epsilon_0 \wedge \neg\Diamond\epsilon_1^1,\\
\epsilon_2^1 & = & \Diamond\epsilon_1^1 \wedge \neg\Diamond\Diamond\epsilon_1^1 \wedge \neg\Diamond\epsilon_1^2, & & \epsilon_2^2 & = & \Diamond\epsilon_1^1 \wedge \Diamond\Diamond\epsilon_1^2 \wedge \neg\Diamond\Diamond\epsilon_1^1,\\
\epsilon_3^1 & = & \Diamond\epsilon_2^1 \wedge \neg\Diamond\Diamond\epsilon_2^1 \wedge \neg\Diamond\epsilon_2^2, & & \epsilon_4^1 & = & \Diamond\epsilon_3^1 \wedge \neg\Diamond\Diamond\epsilon_3^1 \wedge \neg\Diamond\epsilon_2^2,\\
\epsilon_3^2 & = & \Diamond\epsilon_2^2 \wedge \neg\Diamond\Diamond\epsilon_2^2 \wedge \neg\Diamond\epsilon_2^1, & & \epsilon_4^2 & = & \Diamond\epsilon_3^2 \wedge \neg\Diamond\Diamond\epsilon_3^2 \wedge \neg\Diamond\epsilon_2^1.
\end{array}
$$

These variable free formulas characterise points e_j^i in $\mathfrak{F}(\boldsymbol{P}, \langle s, m, n\rangle)$ in the sense that ϵ_j^i is true in $\mathfrak{F}(\boldsymbol{P}, \langle s, m, n\rangle)$ only at the point denoted by e_j^i. Further, let

$$
\begin{array}{rclcrcl}
\phi_0^0 & = & \Diamond\epsilon_2^1 \wedge \Diamond\epsilon_2^2 \wedge \neg\Diamond\Diamond\epsilon_2^1 \wedge \neg\Diamond\Diamond\epsilon_2^2, & & \phi_0^1 & = & \Diamond\epsilon_3^1 \wedge \Diamond\epsilon_3^2 \wedge \neg\Diamond\Diamond\epsilon_3^1 \wedge \neg\Diamond\Diamond\epsilon_3^2,\\
\phi_0^2 & = & \Diamond\epsilon_4^1 \wedge \Diamond\epsilon_4^2 \wedge \neg\Diamond\Diamond\epsilon_4^1 \wedge \neg\Diamond\Diamond\epsilon_4^2, & & \phi_{j+1}^i & = & \Diamond\phi_j^i \wedge \neg\Diamond\Diamond\phi_j^i \wedge \displaystyle\bigwedge_{i\neq k=0}^{2} \neg\Diamond\phi_0^k,
\end{array}
$$

for $i \in \{0, 1, 2\}$, $j \geq 0$. The formula ϕ^i_j is true only at f^i_j. The formulas characterising $s(t, k, l)$ are denoted by $\sigma(t, \phi^1_k, \phi^2_l)$, where

$$\sigma(t, \psi, \chi) \quad = \quad \bigwedge_{i=0}^{t} \Diamond\phi^0_i \wedge \neg\Diamond\phi^0_{t+1} \wedge \Diamond\psi \wedge \neg\Diamond^2\psi \wedge \Diamond\chi \wedge \neg\Diamond^2\chi.$$

We also require formulas characterising not only fixed but arbitrary configurations:

$$\begin{array}{lcl}
\kappa_1 & = & (\Diamond\phi^1_0 \vee \phi^1_0) \wedge \neg\Diamond\phi^0_0 \wedge \neg\Diamond\phi^2_0 \wedge p_1 \wedge \neg\Diamond p_1, \\
\kappa_2 & = & \Diamond\phi^1_0 \wedge \neg\Diamond\phi^0_0 \wedge \neg\Diamond\phi^2_0 \wedge \Diamond p_1 \wedge \neg\Diamond^2 p_1, \\
\pi_1 & = & (\Diamond\phi^2_0 \vee \phi^2_0) \wedge \neg\Diamond\phi^0_0 \wedge \neg\Diamond\phi^1_0 \wedge p_2 \wedge \neg\Diamond p_2, \\
\pi_2 & = & \Diamond\phi^2_0 \wedge \neg\Diamond\phi^0_0 \wedge \neg\Diamond\phi^1_0 \wedge \Diamond p_2 \wedge \neg\Diamond^2 p_2,
\end{array}$$

where p_1 and p_2 are fresh variables.

Now we are fully equipped to simulate the behaviour of Minsky machines by means of modal formulas. Let

$$\blacklozenge\psi \quad = \quad \psi \vee \Diamond\psi \vee \cdots \vee \Diamond^n\psi,$$

where n is a sufficiently large number such that if xR^*y in $\mathfrak{F}(\boldsymbol{P}, \langle s, m, n\rangle)$ then $xR^k y$ for some $k \leq n$. (Note that in the proof of Blok's theorem we took $n = m + 6$.)

With each instruction I in $\boldsymbol{P}$ we associate a formula AxI by taking:

$$AxI \quad = \quad \neg\varphi \wedge \blacklozenge\sigma(t, \pi_1, \kappa_1) \rightarrow \neg\varphi \wedge \blacklozenge\sigma(t', \pi_2, \kappa_1)$$

if I has the form $t \rightarrow \langle t', 1, 0\rangle$,

$$AxI \quad = \quad \neg\varphi \wedge \blacklozenge\sigma(t, \pi_1, \kappa_1) \rightarrow \neg\varphi \wedge \blacklozenge\sigma(t', \pi_1, \kappa_2)$$

if I is $t \rightarrow \langle t', 0, 1\rangle$,

$$\begin{array}{lcl}
AxI & = & (\neg\varphi \wedge \blacklozenge\sigma(t, \pi_2, \kappa_1) \rightarrow \neg\varphi \wedge \blacklozenge\sigma(t', \pi_1, \kappa_1)) \wedge \\
& & \qquad (\neg\varphi \wedge \blacklozenge\sigma(t, \phi^1_0, \kappa_1) \rightarrow \neg\varphi \wedge \blacklozenge\sigma(t'', \phi^1_0, \kappa_1))
\end{array}$$

if I is $t \rightarrow \langle t', -1, 0\rangle\,(\langle t'', 0, 0\rangle)$,

$$\begin{array}{lcl}
AxI & = & (\neg\varphi \wedge \blacklozenge\sigma(t, \pi_1, \kappa_2) \rightarrow \neg\varphi \wedge \blacklozenge\sigma(t', \pi_1, \kappa_1)) \wedge \\
& & \qquad (\neg\varphi \wedge \blacklozenge\sigma(t, \pi_1, \phi^2_0) \rightarrow \neg\varphi \wedge \blacklozenge\sigma(t'', \pi_1, \phi^2_0))
\end{array}$$

if I is $t \rightarrow \langle t', 0, -1\rangle\,(\langle t'', 0, 0\rangle)$. The formula simulating $\boldsymbol{P}$ as a whole is

$$Ax\boldsymbol{P} \quad = \quad \bigwedge_{I \in \boldsymbol{P}} AxI.$$

Now, by induction on the length of computations one can show that, for every program $\boldsymbol{P}$ and all configurations $\langle s, m, n\rangle$, $\langle t, k, l\rangle$, we have the following property ($\Downarrow$):

$$\begin{array}{c}
\boldsymbol{P} : \langle s, m, n\rangle \rightarrow \langle t, k, l\rangle \\
\Downarrow \\
\neg\varphi \wedge \blacklozenge\sigma(s, \phi^1_m, \phi^2_n) \rightarrow \neg\varphi \wedge \blacklozenge\sigma(t, \phi^1_k, \phi^2_l) \quad \in \quad \mathbf{K} \oplus Ax\boldsymbol{P}.
\end{array}$$

Finally, define a logic $L(\boldsymbol{P}, \langle s,m,n\rangle, \langle t,k,l\rangle)$ by taking

$$L(\boldsymbol{P}, \langle s,m,n\rangle, \langle t,k,l\rangle) \;=\; \mathbf{K} \oplus Ax\boldsymbol{P} \oplus (\neg\varphi \wedge \blacklozenge\sigma(s,\phi^1_m,\phi^2_n) \to \neg\varphi \wedge \blacklozenge\sigma(t,\phi^1_k,\phi^2_l)) \to \varphi.$$

Clearly, for given $\boldsymbol{P}$ and configurations $\langle s,m,n\rangle$ and $\langle t,k,l\rangle$, the logic $L(\boldsymbol{P}, \langle s,m,n\rangle, \langle t,k,l\rangle)$ is constructed effectively. We claim that

$$\boldsymbol{P} : \langle s,m,n\rangle \to \langle t,k,l\rangle \qquad \text{iff} \qquad L(\boldsymbol{P}, \langle s,m,n\rangle, \langle t,k,l\rangle) = \mathbf{K} \oplus \varphi.$$

The implication ($\Rightarrow$) is proved using the property ($\Downarrow$) above and the obvious inclusion $L(\boldsymbol{P}, \langle s,m,n\rangle, \langle t,k,l\rangle) \subseteq \mathbf{K} \oplus \varphi$. To show the converse direction, it suffices to observe that if $\boldsymbol{P} : \langle s,m,n\rangle \not\to \langle t,k,l\rangle$ then $\mathfrak{F}(\boldsymbol{P}, \langle s,m,n\rangle)$ validates all the axioms of $L(\boldsymbol{P}, \langle s,m,n\rangle, \langle t,k,l\rangle)$ and refutes φ.

To prove Theorem 9, we modify the definition of the logic $L(\boldsymbol{P}, \langle s,m,n\rangle, \langle t,k,l\rangle)$ above. First we take a formula φ such that $\mathbf{K} \oplus \varphi$ is a consistent tabular logic. One can show that every such logic is not a union-splitting.

Now observe that there exist a program $\boldsymbol{P}$ and a configuration $\langle s,m,n\rangle$ such that no algorithm can decide, given a configuration $\langle t,k,l\rangle$, whether $\boldsymbol{P} : \langle s,m,n\rangle \to \langle t,k,l\rangle$ (for details see [24]). Fix some $\boldsymbol{P}$ and $\langle s,m,n\rangle$ satisfying this condition, and let

$$\begin{aligned} L'(\langle t,k,l\rangle) \quad = \quad & \mathbf{K} \oplus Ax\boldsymbol{P} \oplus \big(\neg\varphi \wedge \blacklozenge\sigma(s,\phi^1_m,\phi^2_n) \to \neg\varphi \wedge \blacklozenge\sigma(t,\phi^1_k,\phi^2_l)\big) \to \varphi \oplus \\ & \neg\varphi \to \bigvee_{i=0}^{l} \Diamond^i\gamma \quad \oplus \quad \neg\varphi \to \bigwedge_{i=0}^{l} \Box^i\big(\gamma \to \Box(\Box_0(\Box_0 p \to p) \to p)\big), \end{aligned}$$

where p is a fresh variable.

If $\boldsymbol{P} : \langle s,m,n\rangle \to \langle t,k,l\rangle$ then, as in the proof of Theorem 7, it is easy to see that we have $L'(\langle t,k,l\rangle) = \mathbf{K} \oplus \varphi$. Thus, this logic is tabular (coincides with the chosen tabular logic, to be more precise), and so it is decidable, Kripke complete and has the fmp.

If $\boldsymbol{P} : \langle s,m,n\rangle \not\to \langle t,k,l\rangle$ then $L'(\langle t,k,l\rangle) \neq \mathbf{K} \oplus \varphi$, which can be shown with the help of the frame $\mathfrak{F}(\boldsymbol{P}, \langle s,m,n\rangle)$. Thus, our logic is different from the chosen tabular logic $\mathbf{K} \oplus \varphi$. Moreover, using the last two axioms (cf. formulas (5), (6) in the proof of Blok's theorem) one can show that although $\varphi \notin L'(\langle t,k,l\rangle)$, no Kripke frame for $L'(\langle t,k,l\rangle)$ can refute φ. It follows that $L'(\langle t,k,l\rangle)$ is Kripke incomplete and does not have the finite model property. Next we use the properties of $\boldsymbol{P}$ and $\langle s,m,n\rangle$ to show that

$$\boldsymbol{P} : \langle s,m,n\rangle \not\to \langle t',k',l'\rangle \quad \text{iff} \quad L'(\langle t,k,l\rangle) \vdash \neg\varphi \wedge \blacklozenge\sigma(s,\phi^1_m,\phi^2_n) \to \neg\varphi \wedge \blacklozenge\sigma(t',\phi^1_{k'},\phi^2_{l'}).$$

The implication ($\Rightarrow$) is proved by induction on the length of computation and ($\Leftarrow$) is shown using the frame $\mathfrak{F}(\boldsymbol{P}, \langle s,m,n\rangle)$. It follows that $L'(\langle t,k,l\rangle)$ is undecidable. ❑

In fact, using the technique above one can prove undecidability of many other important properties of modal logics such as first-order definability (i.e., whether $\mathsf{Fr}\,\{\varphi\}$ is definable by first-order formulas, for an arbitrarily given φ), canonicity, the interpolation and the disjunction properties, etc.; see [24] and references therein. Actually, we know only two interesting decidable properties of finitely axiomatisable logics in NExt $\mathbf{K}$: consistency and coincidence with $\mathbf{K}$. However, even consistency becomes undecidable in NExt $\mathbf{K}_2$ [146].

Theorems 7 and 9 give rise to further interesting questions. First, we still do not know a solution to the following open problem:

PROBLEM 2. Is the set $\{\varphi \in \mathcal{ML}_1 \mid \mathbf{K} \oplus \varphi \text{ is a union-splitting}\}$ decidable?

Second, some of the undecidable problems formulated above may turn out to be *recursively enumerable*, so that we can at least effectively enumerate finitely axiomatisable logics with this or that property. For example, it is easy to show recursive enumerability of the set $\{\varphi \in \mathcal{ML}_n \mid \mathbf{K}_n \oplus \varphi = L\}$, where L is fixed consistent tabular logic (just use the fact that L is finitely axiomatisable, say, by a formula ψ and enumerate those φ from which ψ is derivable in $\mathbf{K}_n$ and vice versa). However, for the majority of important properties of modal logics this problem remains open:

PROBLEM 3. Is it possible to effectively enumerate $\mathcal{ML}_n$-formulas φ for which $\mathbf{K}_n \oplus \varphi$ is decidable (Kripke complete, has the finite model property, interpolation, etc.)?

Finally, one may wonder what happens if we consider the decision problems above for *recursively axiomatisable* modal logics, i.e., those that are given by programs generating their formulas. In this case we have the following analogue of the Rice theorem from general recursion theory:

THEOREM 10 (Kuznetsov). *No nontrivial property of recursively axiomatisable logics is decidable in any of the classes of logics considered above.*

In particular, this result applies even to NExt **S5** (where all logics are finitely axiomatisable)—provided that its logics are represented as programs computing their formulas. Of course, we can recognise, say, consistency in NExt **S5** if all logics in the class are given by finite sets of axioms. The situation is different in, e.g., NExt **S4** where there exist recursively enumerable logics that are not finitely axiomatisable. The proof of this theorem (Kuznetsov left it unpublished) is very simple. In fact, it has nothing to do with modal logics; it is rather about effective computations. The reader can find it in [24].

3.3 Postmortem

So, was the Big Research Programme a failure or a success? Or simply lost illusions?

Dealing with individual systems like $\mathbf{K}_n$, **S4** or even **PDL**, one might think that Modal Logic is 'harmless,' that it is a reasonable compromise between expressiveness and effectiveness, especially in various application areas in computer science and artificial intelligence. Looking at modal logics from a more general perspective, we see, however, that the propositional modal language is extremely expressive, even if we have a single box operator. Modal Logic has been praised by 'users' for being *robustly decidable.* The analysis above shows that, when put in a more general setting, Modal Logic is rather *robustly **un**decidable.* (However, we can take comfort in the mathematical beauty of the splitting-based dichotomy between Kripke completeness and Kripke incompleteness and its transparent repercussions for modal decision problems.)

The outcomes of the Big Research Programme discussed above appear to be similar to the negative solution to the Classical Decision Problem, *das Entscheidungsproblem*, of Hilbert; see [11] and references therein. According to [11], 'the reaction of logicians to the discoveries of Church and Turing was that the classical decision problem was wider than the yes/no version of it . . . The logicians started to think about the classical decision problem as a classification problem. Which fragments are decidable for satisfiability and

which are undecidable? Which fragments are decidable for finite satisfiability and which are undecidable? Which fragments have the finite model property and which contain axioms of infinity (that is satisfiable formulae without finite models)?'

Similar questions make sense in Modal Logic as well. The modal decision problems considered above can be transformed into *modal classification problems*:

(iii) determine (in some sense) maximal classes of modal logics with the desirable properties.

Of course, particularly interesting are natural classes like

- extensions of certain logics, e.g., $\mathbf{K4}$, $\mathbf{S5} \times \mathbf{S5}$, $\mathbf{K4}_t$;
- logics axiomatised by certain 'normal' formulas (e.g., reductions of modalities, Sahlqvist or uniform formulas);
- logics whose classes of frames are closed under certain natural operations (e.g., taking subframes).

To understand the landscape of modal logics in this respect, a variety of different methodologies are required. One established path is to look 'outside' and, e.g., employ modal logics' relation to finite variable/guarded fragments of first-order logic (see Chapter 5), or their relation to languages recognised by tree automata (see Chapter 17). In this chapter we follow the 'internal' approach and analyse how different 'modal' syntactic or semantic restrictions can guarantee this or that desirable property.

4 SYNTACTICAL CLASSES OF MODAL LOGICS

To understand a modal logic is, to a large extent, to understand the structure of its frames, in particular, Kripke frames. An obvious way of doing this is to try to characterise frames by means of first-order formulas in some suitable signature. Classical observations going back to Kripke [79] are as follows, where $\mathfrak{F} = \langle W, R\rangle$ is treated as a Kripke frame in the left-hand column and as a first-order structure in the right-hand one:

$$
\begin{array}{lll}
\mathfrak{F} \models \Box p \to p & \text{iff} & \mathfrak{F} \models \forall x\ R(x,x),\\
\mathfrak{F} \models \Box p \to \Box\Box p & \text{iff} & \mathfrak{F} \models \forall x,y,z\ \big(R(x,y) \land R(y,z) \to R(x,z)\big),\\
\text{etc.} & &
\end{array}
$$

A nice first-order characterisation not only helps in understanding the structure of frames. First, using the standard translation $\cdot^s$ of modal formulas into the first-order language from Section 2 (see also Chapters 1 and 5) and Gödel's completeness theorem, it is readily seen that if $\mathsf{Fr}\,\{\varphi\}$ is definable by a first-order formula as above, then $\mathsf{Log}\,\mathsf{Fr}\,\{\varphi\}$ is recursively enumerable, while in general, by Theorem 1 (a), $\mathsf{Log}\,\mathsf{Fr}\,\{\varphi\}$ might be Π^1_1-hard and even more complex. Second, as was proved by Fine [44] (see also [156, 152]), we have the following:

THEOREM 11 (Fine). *If a logic $L \in \mathrm{NExt}\,\mathbf{K}$ is determined by a first-order definable class of frames then L is $\mathcal{D}$-persistent.*[4]

[4] L is called *$\mathcal{D}$-persistent* if the underlying Kripke frame of any descriptive frame for L validates L as well. A general frame is *descriptive* if it satisfies certain closure conditions which can be found in Chapter 5. If a logic L is $\mathcal{D}$-persistent, then the underlying Kripke frame of its canonical model validates L. In particular, every $\mathcal{D}$-persistent logic is Kripke complete.

This means that to investigate a first-order definable $\mathsf{Log}\,\mathsf{Fr}\,\{\varphi\}$, we can use the well-known methods of canonical models and filtration developed in the 1960–1970s (see, e.g., [24] and references therein). Although the converse of Fine's theorem does not hold, as has been recently shown in [58], it is nevertheless a kind of empirical rule that logics not determined by first-order definable classes are *not* $\mathcal{D}$-persistent, and therefore, the standard way of proving completeness or the finite model property is blocked for them.

By Chagrova's theorem [29], there is no effective way of deciding, given a formula φ, whether $\mathsf{Fr}\,\{\varphi\}$ is first-order definable. However, one can try to find and describe *syntactically* some classes of formulas φ for which $\mathsf{Fr}\,\{\varphi\}$ is first-order definable. In fact, this approach has been the driving force behind much research in Modal Logic since the 1960s (see, e.g., [88]). The (so far) most general syntactically defined class of formulas for which this holds true was discovered by H. Sahlqvist [125].

4.1 Sahlqvist logics

Sahlqvist's theorem [125] (see also [53])—perhaps the most celebrated general result in Modal Logic—gives a sufficient condition for first-order definability and Kripke completeness of logics in $\mathrm{NExt}\,\mathbf{K}_n$. To formulate it we require the following definitions.

Say that a formula is *positive* if it is constructed from variables and the constants $\top$, $\bot$ using $\wedge$, $\vee$, $\Diamond_i$ and $\Box_i$. An arbitrary finite sequence of boxes $\Box_i$, $i = 1, \dots, n$ will be denoted by $\Box^*$.

A formula $\varphi \in \mathcal{ML}_n$ is called a *Sahlqvist formula* if it is equivalent in $\mathbf{K}_n$ to a formula of the form $\Box^*(\psi \to \chi)$, where χ is positive and ψ is constructed from variables and their negations, $\bot$ and $\top$ with the help of $\wedge$, $\vee$, $\Box_i$ and $\Diamond_i$ in such a way that no subformula of ψ of the form $\psi_1 \vee \psi_2$ or $\Diamond_i \psi_1$, containing an occurrence of a variable without $\neg$, is in the scope of some $\Box_j$. For example, formulas (2)–(3) from Section 2 are Sahlqvist, while the Löb axiom (1) and the McKinsey axiom

$$\boldsymbol{ma} \quad = \quad \Box\Diamond p \to \Diamond\Box p \tag{7}$$

are not.

THEOREM 12 (Sahlqvist). (a) *Given a Sahlqvist formula $\varphi \in \mathcal{ML}_n$, one can effectively construct a first-order formula $\phi(x)$ in $R_1, \dots, R_n$ and $=$ having x as its only free variable and such that, for every descriptive or Kripke frame $\mathfrak{F}$ and every point a in $\mathfrak{F}$,*

$$(\mathfrak{F}, a) \models \varphi \qquad \textit{iff} \qquad \mathfrak{F} \models \phi(x)[a].$$

(*Here $(\mathfrak{F}, a) \models \varphi$ means that φ is true at a in $\mathfrak{F}$ under any valuation.*)

(b) *If Γ is a set of Sahlqvist $\mathcal{ML}_n$-formulas and $L \in \mathrm{NExt}\,\mathbf{K}_n$ is a $\mathcal{D}$-persistent logic then the logic $L \oplus \Gamma$ (in particular, $\mathbf{K}_n \oplus \Gamma$) is $\mathcal{D}$-persistent as well. Moreover, $L \oplus \Gamma$ is elementary (in the sense that the class of Kripke frames for it coincides with the class of all models for some set of first-order formulas in R_i and $=$) whenever L is so.*

Various detailed proofs of this result can be found in [126, 70, 7] (for some generalisations see, e.g., [30, 60, 72, 50]). So, instead of going into technical details, we will concentrate on the meaning of Sahlqvist's theorem.

First, it gives much more than just first-order definability of $\mathsf{Fr}\,\{\varphi\}$, for a Sahlqvist formula φ, and, therefore, recursive enumerability of $\mathsf{Log}\,\mathsf{Fr}\,\{\varphi\}$. In fact, we also obtain an axiomatisation, namely, that $\mathsf{Log}\,\mathsf{Fr}\,\{\varphi\} = \mathbf{K} \oplus \varphi$. As we know from Blok's theorem,

this is much stronger than just first-order definability. (For example, there are a lot of formulas φ such that $\mathsf{Fr}\{\varphi\}$ is the class of transitive frames, but $\mathbf{K} \oplus \varphi \neq \mathbf{K4}$.) Thus, Sahlqvist's theorem has two aspects: the correspondence part (stating first-order definability of $\mathsf{Fr}\{\varphi\}$) and the completeness part (stating that $\mathsf{Log}\,\mathsf{Fr}\{\varphi\} = \mathbf{K} \oplus \varphi$).

However, Sahlqvist axioms do not guarantee good computational properties of modal logics. For example, there are finitely axiomatisable Sahlqvist logics without the finite model property in NExt $\mathbf{S4}$ [26] (see also [65]). There are undecidable finitely axiomatisable Sahlqvist logics in NExt $\mathbf{K}$. Such a logic can easily be constructed if we have more than one box [23]. For instance, consider the undecidable associative calculus T of [148] with the axioms

$$ac = ca, \quad ad = da, \quad bc = cb, \quad bd = db, \quad edb = be, \quad eca = ae, \quad abac = abacc.$$

The reader will notice immediately an analogy between these axioms and the axioms of the following modal logic with five necessity operators:

$$\begin{aligned} L \quad = \quad \mathbf{K}_5 \quad \oplus \quad & \Box_1\Box_3 p \leftrightarrow \Box_3\Box_1 p \ \oplus \ \Box_1\Box_4 p \leftrightarrow \Box_4\Box_1 p \ \oplus \ \Box_2\Box_3 p \leftrightarrow \Box_3\Box_2 p \ \oplus \\ & \Box_2\Box_4 p \leftrightarrow \Box_4\Box_2 p \ \oplus \Box_5\Box_4\Box_2 p \leftrightarrow \Box_2\Box_5 p \ \oplus \ \Box_5\Box_3\Box_1 p \leftrightarrow \Box_1\Box_5 p \ \oplus \\ & \Box_1\Box_2\Box_1\Box_3 p \leftrightarrow \Box_1\Box_2\Box_1\Box_3\Box_3 p. \end{aligned}$$

Moreover, it is not hard to see that words x, y in the alphabet $\{a, b, c, d, e\}$ are equivalent in T iff $f(x)p \leftrightarrow f(y)p \in \mathbf{K}_5$, where f is the natural one-to-one correspondence between such words and modalities in language $\{\Box_1, \dots, \Box_5\}$ under which, for instance, $f(cadedb) = \Box_3\Box_1\Box_4\Box_5\Box_4\Box_2$. It follows immediately that the Sahlqvist 5-modal logic L is undecidable. An even simpler example of an undecidable finitely axiomatisable Sahlqvist logic is the bimodal product $\mathbf{K4} \times \mathbf{K4}$; for details see Chapter 15. Now, using the reduction of multi-modal logics to those in NExt $\mathbf{K}$ [77] one can construct an undecidable finitely axiomatisable Sahlqvist logic from NExt $\mathbf{K}$.

PROBLEM 4. Are finitely axiomatisable Sahlqvist logics in NExt $\mathbf{K4}$ decidable?

It is also worth noting that there is no effective way of recognising whether a given modal formula is (deductively equivalent to) a Sahlqvist formula; in particular, the set

$$\{\varphi \in \mathcal{ML}_1 \mid \mathbf{S4} \oplus \varphi \text{ is a Sahlqvist logic}\}$$

is not recursive [26].

The simplest formula not covered by Sahlqvist's theorem is the Mckinsey axiom $\boldsymbol{ma}$ (see (7) above). It is neither first-order definable [56, 154][5], nor canonical [55]. The problem whether the following equality holds

$$\mathbf{K} \quad \oplus \quad \Box\Diamond p \to \Diamond\Box p \quad = \quad \mathsf{Log}\,\mathsf{Fr}\{\Box\Diamond p \to \Diamond\Box p\}$$

and whether the logic $\mathbf{K} \oplus \boldsymbol{ma}$ is decidable had resisted all attempts based on the standard methods of canonical models and filtration until Fine [44] introduced a new proof technique based on certain normal forms to be considered in the next section.

Another logic not covered by Sahlqvist's theorem is $\mathbf{KM}^\infty = \mathbf{K} \oplus \{\boldsymbol{ma}_k \mid k \geq 1\}$ defined in [88], where

$$\boldsymbol{ma}_k \quad = \quad \Diamond \bigwedge_{1 \leq i \leq k} (\Diamond p_i \to \Box p_i).$$

[5]This result was first proved by R. Goldblatt in his PhD thesis in 1974.

$\boldsymbol{ma}_1$ is $\mathbf{K}$-equivalent to $\boldsymbol{ma}$, so $\mathbf{KM}^\infty \supseteq \mathbf{K} \oplus \boldsymbol{ma}$. In fact, $\mathbf{KM}^\infty$ is the logic of the class of frames satisfying

$$\forall x \exists y \ \big(R(x,y) \wedge \forall z, z' \ (R(y,z) \wedge R(y,z') \to z = z')\big), \tag{8}$$

so by Theorem 11 it is canonical.

In [64], the proof of Sahlqvist's theorem is extended to $\mathbf{KM}^\infty$ and other logics. The method uses *'quasipositive' hybrid sentences*; see Chapter 14 for full details of hybrid logic. In these formulas, existential and relativised universal quantifiers over nominals are allowed, negation can only occur in the latter, and there are no free nominals or propositional variables. From any quasipositive sentence φ, an infinite set of *modal* axioms can be obtained effectively. The modal axioms approximate φ, by treating nominals as propositional variables ranging over the partition sets of finite partitions of the worlds of a model. Each partition is defined by the truth values of an arbitrary finite set of modal formulas. Existential and universal quantification are simulated by disjunctions and conjunctions over partition sets. The axioms obtained in this way axiomatise a modal logic L_φ, which is shown to be the logic of the class of frames validating φ: i.e., $L_\varphi = \mathsf{Log}\,\mathsf{Fr}\,\{\varphi\}$.

For example, let $\varphi = \Diamond \exists i \Box i$, where i is a nominal. Then φ is valid in precisely the frames satisfying (8). The axioms obtained from φ are equivalent to substitution instances of the $\boldsymbol{ma}_k$ above, and so $L_\varphi = \mathbf{KM}^\infty$. For instance, the axiom obtained by approximating φ with respect to the finite set $X = \{p_1, \dots, p_k\}$ is

$$\Diamond \bigvee_{Y \subseteq X} \Box \Big(\bigwedge_{p \in Y} p \wedge \bigwedge_{p \in X - Y} \neg p \Big),$$

and this is $\mathbf{K}$-equivalent to $\boldsymbol{ma}_k$.

The method extends to sets Φ of quasipositive sentences. Every L_Φ is the logic of an elementary class of frames, namely, $\mathsf{Fr}\,\Phi$. [64] shows that the modal logics of elementary classes of frames are *precisely* those of the form Λ_Φ. The result applies to multi-modal logics and to logics with polyadic modalities.

This result gives an interesting link between modal and hybrid logic. It is analogous to Sahlqvist's completeness theorem, since $\mathsf{Log}\,\mathsf{Fr}\,\Phi = L_\Phi$. An analogue of Sahlqvist's correspondence theorem would state that $\mathsf{Fr}\,L_\Phi$ is first-order definable (by Φ), but this cannot be achieved in general, since in many cases, $\mathsf{Fr}\,L_\Phi$ is non-elementary.

4.2 Uniform logics

Fine [44] used a modal analogue of the full disjunctive normal form for constructing finite models and proving the fmp of a family of logics in NExt $\mathbf{D}$ (containing, in particular, $\mathbf{K} \oplus \boldsymbol{ma}$).

Observe first that every modal formula $\varphi(p_1, \dots, p_m)$ is equivalent in $\mathbf{K}$ either to $\bot$ or to a disjunction of normal forms (in the variables $p_1, \dots, p_m$) of degree $md(\varphi)$ (the modal depth of φ), which are defined inductively in the following way. $\mathbf{NF}_0$, the set of *normal forms of degree* 0, contains all formulas of the form $\neg_1 p_1 \wedge \cdots \wedge \neg_m p_m$, where each $\neg_i$ is either blank or $\neg$. $\mathbf{NF}_{n+1}$, the set of *normal forms of degree* $n+1$, consists of formulas of the form

$$\theta \wedge \neg_1 \Diamond \theta_1 \wedge \cdots \wedge \neg_k \Diamond \theta_k,$$

where $\theta \in \mathbf{NF}_0$ and $\theta_1, \ldots, \theta_k$ are all distinct normal forms in $\mathbf{NF}_n$. Put $\mathbf{NF} = \bigcup_{n<\omega} \mathbf{NF}_n$. Using the fact that $\bigvee\{\Diamond\theta \mid \theta \in \mathbf{NF}_n\} \in \mathbf{D}$, it is not hard to see also that in $\mathbf{D}$ every formula φ with $md(\varphi) \leq n$ is equivalent either to $\bot$ or to a disjunction of normal forms of degree n such that at least one of $\neg_1, \ldots, \neg_k$ in the inductive step of the definition above is blank. Such normal forms are called $\mathbf{D}$-*suitable*. It should be clear that, for any distinct $\theta', \theta'' \in \mathbf{NF}_n$, $\neg(\theta' \wedge \theta'') \in \mathbf{K}$. Consequently, for every $\theta \in \mathbf{NF}_n$ and every $\varphi(p_1, \ldots, p_m)$ with $md(\varphi) \leq n$, we have either $\theta \to \varphi \in \mathbf{K}$ or $\theta \to \neg\varphi \in \mathbf{K}$.

With each $\mathbf{D}$-suitable normal form θ we associate a model $\mathfrak{M}_\theta = \langle \mathfrak{F}_\theta, \mathfrak{V}_\theta \rangle$ based on $\mathfrak{F}_\theta = \langle W_\theta, R_\theta \rangle$ by taking

$$\begin{array}{lll} W_\theta & = & \{\top\} \cup \{\theta' \in \mathbf{NF} \mid \theta' <^n \theta, \text{ for some } n \geq 0\}, \\ \theta' < \theta'' & \text{iff} & \Diamond\theta' \text{ is a conjunct of } \theta'', \\ \theta' R_\theta \theta'' & \text{iff} & \text{either } \theta' > \theta'' \text{ or } md(\theta') = 0 \text{ and } \theta'' = \top, \\ \mathfrak{V}_\theta(p) & = & \{\theta' \in W_\theta \mid p \text{ is a conjunct of } \theta'\}. \end{array}$$

According to the definition, $\top$ is the reflexive end-point in $\mathfrak{F}_\theta$, and so $\mathfrak{F}_\theta$ is serial. By a straightforward induction on the degree of $\theta' \in W_\theta$ one can show that $(\mathfrak{M}_\theta, \theta') \models \theta'$. It follows immediately that $\mathbf{D}$ has the finite model property (fmp, for short). Indeed, given $\varphi \notin \mathbf{D}$, we reduce $\neg\varphi$ to a disjunction of $\mathbf{D}$-suitable normal forms with at least one disjunct θ, and then $(\mathfrak{M}_\theta, \theta) \models \theta$.

It turns out that in the same way we can prove the fmp of all logics in NExt $\mathbf{D}$ that are axiomatisable by *uniform formulas* which are defined as follows. Every φ without modal operators is a *uniform formula of degree* 0; and if $\varphi = \psi(\blacklozenge_1\chi_1, \ldots, \blacklozenge_m\chi_m)$, where $\blacklozenge_i \in \{\Box, \Diamond\}$, $md(\psi(p_1, \ldots, p_m)) = 0$ and $\chi_1, \ldots, \chi_m$ are uniform formulas of degree n, then φ is a *uniform formula of degree* $n+1$. A remarkable property of uniform formulas is the following. Suppose that φ is a uniform formula of degree n and $\mathfrak{M}$, $\mathfrak{N}$ are models based on the same frame $\mathfrak{F} = \langle W, R \rangle$ and such that, for some point x, $(\mathfrak{M}, y) \models p$ iff $(\mathfrak{N}, y) \models p$ for every $y \in x{\uparrow}^n$ and every variable p in φ, where

$$X{\uparrow} \;=\; X{\uparrow}^1 \;=\; \{y \in W \mid \exists x \in X\; xRy\}, \qquad X{\uparrow}^{n+1} \;=\; (X{\uparrow}^n){\uparrow}\,.$$

Then $(\mathfrak{M}, x) \models \varphi$ iff $(\mathfrak{N}, x) \models \varphi$.

Given a logic L, we call a normal form θ L-*suitable* if $\mathfrak{F}_\theta \models L$.

THEOREM 13 (Fine). *Every logic $L \in$ NExt$\mathbf{D}$ axiomatisable by uniform formulas has the fmp.*

Proof. It suffices to prove that each formula φ with $md(\varphi) \leq n$ is equivalent in L either to $\bot$ or to a disjunction of L-suitable normal forms of degree n. And this fact will be established if we show that every $\mathbf{D}$-suitable normal form θ such that $\theta \to \bot \notin L$ is L-suitable. Suppose otherwise. Let θ be an L-consistent and $\mathbf{D}$-suitable normal form of the least possible degree under which it is not L-suitable. Then there are a uniform formula $\psi \in L$ of some degree m and a model $\mathfrak{M} = \langle \mathfrak{F}_\theta, \mathfrak{V} \rangle$ such that $(\mathfrak{M}, \theta) \not\models \psi$.

For every variable p in ψ, let $\Gamma_p = \{\theta' \in \theta{\uparrow}^m \mid (\mathfrak{M}, \theta') \models p\}$ and let $\delta_p = \bigvee \Gamma_p$ (if $\Gamma_p = \emptyset$ then $\delta_p = \bot$). Observe that for every $\theta' \in \theta{\uparrow}^m$ we have $(\mathfrak{M}_\theta, \theta') \models \delta_p$ iff $\theta' \in \Gamma_p$ iff $(\mathfrak{M}, \theta') \models p$. Therefore, the formula ψ' that results from ψ by replacing each p with δ_p is false at θ in $\mathfrak{M}_\theta$. Now, if $md(\psi') > n$ then $m > n$, and so $\delta_p = \bot$ for every p in ψ, i.e., ψ' is variable free. But then ψ' is equivalent in $\mathbf{D}$ to $\top$ or $\bot$, contrary to $\mathfrak{F}_\theta \not\models \psi'$ and L being consistent. And if $md(\psi') \leq n$ then either $\theta \to \psi' \in \mathbf{K}$, which is impossible,

since $(\mathfrak{M}_\theta, \theta) \not\models \theta \to \psi'$, or $\theta \to \neg\psi' \in \mathbf{K}$, from which $\psi' \to \neg\theta \in \mathbf{K}$ and so $\neg\theta \in L$, contrary to θ being L-consistent. ❑

It is not hard to extend Fine's theorem to the multi-modal case, namely, to those logics that contain $\Diamond_i \top$, for all $i = 1, \dots, n$, and are axiomatisable by formulas φ in which all maximal sequences of nested modal operators coincide with respect to the distribution of the indices i of $\Box_i$ and $\Diamond_i$.

As a consequence of Theorem 13 we obtain that $\mathbf{KM} = \mathbf{K} \oplus \boldsymbol{ma}$ enjoys the fmp and so is decidable. Strange as it may seem, the following problem is still open:

PROBLEM 5. What is the computational complexity of **KM**?

4.3 Logics with $\Box\Diamond$-axioms

Another result, connecting the fmp of logics with the distribution of $\Box$ and $\Diamond$ over their axioms, is based on the following observation which can be regarded as the modal analogue of Glivenko's theorem for intuitionistic logic (see, e.g., [24]): for all formulas $\varphi, \psi \in \mathcal{ML}_1$, we have $\Diamond\varphi \leftrightarrow \Diamond\psi \in \mathbf{S5}$ iff $\Box\Diamond\varphi \leftrightarrow \Box\Diamond\psi \in \mathbf{K4}$. The proof of this observation is almost trivial. Suppose $\Box\Diamond\varphi \to \Box\Diamond\psi \notin \mathbf{K4}$. Then there exist a finite model $\mathfrak{M}$, based on a transitive frame, and a point x in it such that $x \models \Box\Diamond\varphi$ and $x \not\models \Box\Diamond\psi$. It follows from the former that every final cluster accessible from x, if any, is non-degenerate and contains a point where φ is true. The latter means that x 'sees' a final cluster C at all points of which ψ is false. Now, by taking the generated submodel of $\mathfrak{M}$ based on C, we obtain a model for **S5** refuting $\Diamond\varphi \to \Diamond\psi$. The rest is obvious, since $\Diamond p \leftrightarrow \Box\Diamond p$ is in **S5** and $\mathbf{K4} \subseteq \mathbf{S5}$.

$\mathcal{ML}_1$-formulas in which every occurrence of a variable is in the scope of a modality $\Box\Diamond$ will be called *$\Box\Diamond$-formulas*. The next theorem is due to Rybakov [124].

THEOREM 14 (Rybakov). *If a logic $L \in \mathrm{NExt}\,\mathbf{K4}$ is decidable* (*or has the fmp*) *and ψ is a $\Box\Diamond$-formula then $L \oplus \psi$ is also decidable* (*has the fmp*).

Proof. Suppose that $\psi = \psi'(\Box\Diamond\chi_1, \dots, \Box\Diamond\chi_n)$, for some formula $\psi'(q_1, \dots, q_n)$. If $\varphi(p_1, \dots, p_m) \in L \oplus \psi$ then there exists a derivation of φ in $L \oplus \psi$ in which substitution instances of ψ contain no variables different from $p_1, \dots, p_m$. Each of these instances has the form $\psi'(\Box\Diamond\chi'_1, \dots, \Box\Diamond\chi'_n)$, where every χ'_i is some substitution instance of χ_i containing only $p_1, \dots, p_m$. Now, it is not hard to see that, similarly to classical propositional logic, there are finitely many pairwise nonequivalent formulas in **S5** built from $p_1, \dots, p_m$ (for more details see, e.g., [129] or [24]). In view of the observation above, there are finitely many pairwise nonequivalent in **K4** substitution instances of $\Box\Diamond\chi_i$ of that sort (the reader can easily estimate the number of them). So there exist only finitely many pairwise nonequivalent in **K4** substitution instances of ψ containing $p_1, \dots, p_m$, say, $\psi_1, \dots, \psi_k$, and we can effectively construct them. Then, by the deduction theorem,

$$\varphi \in L \oplus \psi \qquad \text{iff} \qquad \Box^+(\psi_1 \wedge \dots \wedge \psi_k) \to \varphi \in L,$$

where $\Box^+\chi = \chi \wedge \Box\chi$. Thus, $L \oplus \psi$ is decidable (or has the fmp) whenever L is decidable (has the fmp). ❑

It should be noted that by adding infinitely many $\Box\Diamond$-formulas to a logic L with the fmp one can construct a Kripke incomplete logic; for a concrete example see [123].

4.4 Logics with noniterative axioms

Lewis [91] considered those logics in NExt $\mathbf{K}_n$ that can be axiomatised by $\mathcal{ML}_n$-formulas without nested modal operators. We call such logics *noniterative.* Examples of noniterative logics are

$$\mathbf{T} \;=\; \mathbf{K} \oplus \Box p \to p \qquad \text{or} \qquad \mathbf{K}_2 \oplus \Box_2 p \to \Box_1 p.$$

THEOREM 15 (Lewis). *All noniterative logics in* NExt $\mathbf{K}_n$ *have the fmp.*

Proof. Suppose that the axioms of $L = \mathbf{K}_n \oplus \Gamma$ have no nested modal operators and $\varphi \notin L$. Let $\mathsf{sub}\,\varphi$ be the set of all subformulas of φ. By a *φ-description* we mean any set of subformulas of φ together with the negations of the remaining formulas in $\mathsf{sub}\,\varphi$. For each L-consistent φ-description Θ, take a maximal L-consistent set Δ_Θ containing Θ. Denote by W the (finite) set of the selected Δ_Θ and define $\mathfrak{F} = \langle W, (R_i \mid i \in I)\rangle$ and $\mathfrak{M} = \langle \mathfrak{F}, \mathfrak{V}\rangle$ by taking

$$\Delta_\Theta R_i \Delta_\Psi \qquad \text{iff} \qquad \Diamond_i \bigwedge \Psi \in \Delta_\Theta$$

and $\mathfrak{V}(p) = \{\Delta_\Theta \in W \mid p \in \Delta_\Theta\}$. It is easily proved that $(\mathfrak{M}, \Delta_\Theta) \models \psi$ iff $\psi \in \Delta_\Theta$, for all subformulas ψ of φ and $\Delta_\Theta \in W$. Hence $\mathfrak{F} \not\models \varphi$. It is also easy to see that for all truth-functional compounds ψ of subformulas of φ,

(9) $\quad (\mathfrak{M}, \Delta_\Theta) \models \Diamond_i \psi \qquad \text{iff} \qquad \Diamond_i \psi \in \Delta_\Theta.$

Consider now a model $\mathfrak{M}' = \langle \mathfrak{F}, \mathfrak{V}'\rangle$ and $\chi \in \Gamma$. For each variable p put

$$\psi_p \quad = \quad \bigvee \left\{ \bigwedge \Theta \mid \Delta_\Theta \in \mathfrak{V}'(p) \right\}$$

and denote by χ' the result of substituting ψ_p for p, for each p in χ. Then $\mathfrak{M}' \models \chi$ iff $\mathfrak{M} \models \chi'$. In view of (9), we have $\mathfrak{M} \models \chi'$ because χ' has no nested modalities. Thus, $\mathfrak{F} \models \chi$ and so $\mathfrak{F} \models L$. ❑

4.5 Modal reduction principles

Modal reduction principles—that is, formulas of the form $\boldsymbol{M}p \to \boldsymbol{N}p$, where $\boldsymbol{M}$ and $\boldsymbol{N}$ are strings of $\Box_i$ and $\Diamond_j$—have always attracted the attention of modal logicians, with the aim being to reduce the number of nested modal operators. (For example, both $\Box\Diamond p \leftrightarrow \Diamond p$ and $\Diamond\Box p \leftrightarrow \Box p$ are in $\mathbf{S5}$.) In the context of this chapter, we are interested in completeness and decidability of normal modal logics axiomatised by modal reduction principles.

As we know from Section 4.1, there are undecidable logics in NExt $\mathbf{K}_2$ with finitely many modal reduction principles as their axioms. But it seems that nearly nothing is known about the behaviour of logics with such axioms in NExt $\mathbf{K}$. Perhaps one of the most intriguing open problems in Modal Logic is the following:

PROBLEM 6. Do the logics of the form $\mathbf{K} \oplus \Box^n p \to \Box^m p$ have the finite model property?

Note, however, that by Sahlqvist theorem all logics of the form $\mathbf{K} \oplus \Box^n p \to \Box^m p$ are characterised by their Kripke frames that are definable by first-order formulas (which are

similar to transitivity: if y is accessible from x in m steps, then y is accessible from x in n steps as well).

Van Benthem [155] showed that modal reduction principles in $\mathcal{ML}_1$ are all first-order definable over *transitive* frames (this is not the case in general: e.g., the McKinsey axiom $\Box\Diamond p \to \Diamond\Box p$ is not first-order definable over arbitrary frames; see [155] for a complete characterisation).

The following result was proved in [175] using the method of canonical formulas to be discussed in Section 6:

THEOREM 16. *All logics in* NExt **K4** *axiomatisable by modal reduction principles have the fmp and are decidable.*

PROBLEM 7. Are extensions of $\mathbf{K}_n$ with modal reduction principles Kripke complete?

PROBLEM 8. Are extensions of **K** with modal reduction principles decidable?

4.6 Logics with n-variable axioms

A very natural syntactical parameter of a modal logic $L \oplus \Gamma$ is the number of variables in its extra axioms Γ over L. For example, $\mathbf{D} = \mathbf{K} \oplus \Diamond\top$ is axiomatised by a *variable-free* formula over **K** and almost all standard modal logics—e.g., **K4**, **S4**, **S5**, **Grz**, **GL**—can be axiomatised by adding axioms with only one variable to **K**. (A notable exception is **K4.3** whose axiomatisation requires two variables; see [119]).

We start our discussion of variable-free axioms with a simple observation that the truth of a variable-free formula φ does not depend on the valuation, i.e., for every model $\mathfrak{M}$ based on a frame $\mathfrak{F}$, we have $\mathfrak{M} \models \varphi$ iff $\mathfrak{F} \models \varphi$. Therefore, we can reduce deduction in $L \oplus \varphi$ to 'global' deduction in L. More precisely, we say that a formula ψ *follows globally* from a set Γ of formulas in a logic L if $\mathfrak{M} \models \Gamma$ implies $\mathfrak{M} \models \psi$ for every model $\mathfrak{M}$ based on a frame for L. Now, if global deduction is decidable for L, then $L \oplus \varphi$ is decidable: indeed, $\psi \in L \oplus \varphi$ iff ψ follows globally from φ in L.

To make use of this observation we need to know how to prove decidability of 'global deducibility' for modal logics L. For $L \in$ NExt **K4** this is simple because ψ follows globally from φ in L iff $\varphi \wedge \Box\varphi \to \psi \in L$.

THEOREM 17. *If φ is variable-free and $L \in$ NExt* **K4** *is decidable, then $L \oplus \varphi$ is decidable.*

Note that there are extensions of **K4** with *infinitely many* variable free axioms which are undecidable and do not have the fmp; for a concrete example and further details see [24]. This cannot happen in NExt **GL**, however, because each variable-free formula is deductively equivalent in **GL** to one of the formulas $\top$, $\Box^n\bot$, where $n < \omega$. Since $\Box^i\bot \to \Box^j\bot \in \mathbf{K4} \subseteq \mathbf{GL}$, for $i \leq j$, all extensions of **GL** with variable-free formulas are finitely axiomatisable and decidable.

The simple reduction of global deducibility above does not apparently work for 'non-transitive' logics in NExt **K**. In fact, there are decidable normal modal logics L such that global deducibility is undecidable for L. (The first example of such a logic was constructed by Spaan [139]. Another natural example is $\mathbf{K} \times \mathbf{K}$; for details see Chapter 15 or [48].) It should be clear that global deducibility in a finitely axiomatisable modal logic L is decidable if L enjoys the so-called *global finite model property* (global fmp, for short): for every finite set $\Gamma \cup \{\psi\}$ of formulas, ψ does not follow globally from Γ iff there exists a finite model $\mathfrak{M}$ based on a frame for L such that $\mathfrak{M} \models \Gamma$ and $\mathfrak{M} \not\models \varphi$. The global

fmp of many standard modal logics like $\mathbf{K}_n$, $\mathbf{K4}_n$, $\mathbf{S5}_n$ can be proved by filtration: just start with a possibly infinite model $\mathfrak{M}$ such that $\mathfrak{M} \models \Gamma$ and $\mathfrak{M} \not\models \psi$, and then filtrate it through the subformulas of $\Gamma \cup \{\psi\}$; see [59] for details. For example, we have the following:

THEOREM 18. *If φ is variable-free, then $\mathbf{K}_n \oplus \varphi$ is decidable.*

The undecidable Sahlqvist logic on page 447 shows that this result does not hold for axioms with one variable. Chagrov [21] constructed a one-variable formula φ such that $\mathbf{GL} \oplus \varphi$ is undecidable. It turns out, however, that we have the following theorem which was proved in [175] using the method of canonical formulas (to be discussed in Section 6):

THEOREM 19. *For every one-variable formula φ, $\mathbf{S4} \oplus \varphi$ has the fmp and is decidable.*

On the other hand, an infinite number of one-variable axioms can yield an extension of **S4** without the fmp [133]. A Kripke incomplete extension of **S4** with a two-variable axiom was constructed by Shehtman [132] and an undecidable logic above **S4** with a three-variable axiom by Chagrov [21].

PROBLEM 9. Are extensions of **S4** with a two-variable axiom decidable?

5 SEMANTICALLY CONSTRAINED CLASSES OF MODAL LOGICS

In the previous section we considered 'nice' classes of modal logics defined in terms of the form of the logics' axioms. Here we give a brief overview of well-behaved classes of modal logics determined by imposing some natural constraints on the form of their frames.

Tabular logics

A normal modal logic L is said to be *tabular* if it is determined by a *finite* set of *finite* frames . Since the class of frames for a normal modal logic is closed under disjoint unions, L is tabular iff there exists a single finite frame that determines L. In many respects tabular logics are easy to deal with. For instance, the problem of deciding whether a formula φ belongs to a tabular logic is trivially decided in NP by considering all possible valuations in the finite frame characterising L. Moreover, it is not difficult to provide a finite axiomatisation for a tabular logic; for details see, e.g., [177]. Thus, we arrive at the following:

THEOREM 20. *Every tabular logic is coNP-complete and finitely axiomatizable. Moreover, a normal modal logic is tabular if, and only if, it contains one of the formulas*

$$\boldsymbol{tab}_n \;=\; \neg(\varphi_1 \wedge \Diamond(\varphi_2 \wedge \Diamond(\varphi_3 \wedge \cdots \wedge \Diamond\varphi_n)\dots)) \wedge \bigwedge_{m=0}^{n-1} \neg\Diamond^m(\Diamond\varphi_1 \wedge \dots \wedge \Diamond\varphi_n)$$

where $\varphi_i = p_1 \wedge \cdots \wedge p_{i-1} \wedge \neg p_i \wedge p_{i+1} \wedge \cdots \wedge p_n$.

What is the position of tabular logics within the lattices NExt $\mathbf{K}_n$? First, it is easy to see that every normal modal logic containing a tabular modal logic L is tabular as well and is determined by frames that are p-morphic images of generated subframes of any frame which determines L. Therefore, we have:

THEOREM 21. *If L is tabular then* NExt L *is finite and contains only tabular logics.* NExt L *can be effectively computed.*

On the other hand, according to Theorem 9, the axiomatisation problem for tabular modal logics is always undecidable in NExt $\mathbf{K}_n$. The situation is not so hopeless if we consider the following relativised version of the axiomatisation problem for tabular logics above some sufficiently 'strong' logic $L_0 \supset \mathbf{K}_n$: given a tabular logic $L \supset L_0$ and an arbitrary formula φ, decide whether $L_0 \oplus \varphi = L$. For example, one can easily show (see, e.g., [24, 177] and references therein) that every tabular logic containing **K4** is a union-splitting of **K4** and that a logic is tabular in NExt **K4** iff it has finitely many normal extensions. Moreover, the following holds:

THEOREM 22. *If $L \in$ NExt* **K4** *is tabular then $\{\varphi \mid \mathbf{K4} \oplus \varphi = L\}$ is decidable.*

How to determine whether a given logic is tabular? The key idea suggested by Kuznetsov [82] is to consider the so-called pretabular logics.

A logic $L \in \text{NExt}\, L_0$ is said to be *pretabular* if L is not tabular but every proper extension of L in NExt L_0 is tabular. In other words, a pretabular logic is a maximal nontabular logic in NExt L_0. Using Zorn's lemma it is easily seen that, in NExt $\mathbf{K}_n$, every non-tabular logic is contained in a pretabular one. It is also known that every pretabular logic in NExt **K4** has the fmp (for proofs and references consult [24]). Moreover, Maksimova [96] and Esakia and Meskhi [38] showed that there are only five (pretty simple) pretabular logics in NExt **S4**. Using this result one can show the following:

THEOREM 23. *The set $\{\varphi \mid \mathbf{S4} \oplus \varphi$ is tabular$\}$ is decidable.*

Indeed, we launch two parallel processes: one of them generates all derivations in $\mathbf{S4} \oplus \varphi$ and stops after finding a derivation of $\boldsymbol{tab}_n$, for some $n < \omega$; another process checks if φ belongs to a pretabular logic in NExt **S4** and stops if this is the case. The termination of the first process means that $\mathbf{S4} \oplus \varphi$ is tabular, and if the second one comes to a stop then this logic is not tabular.

Note that there are a continuum of pretabular logics in NExt **K4**, while NExt **GL** contains countably many of them [10, 17], and the set $\{\varphi \mid \mathbf{GL} \oplus \varphi$ is tabular$\}$ is decidable.

PROBLEM 10. Is the set $\{\varphi \mid \mathbf{K4} \oplus \varphi$ is tabular$\}$ decidable?

Transitive logics of finite depth and width

A very natural semantical constraint on logics from NExt **K4** is the length of maximal chains and antichains in their rooted frames. Say that a logic $L \in$ NExt **K4** is of *depth* $n < \omega$ if L has a frame $\langle W, R\rangle$ with a chain $x_1 R x_2 R \dots R x_n$ of points from distinct clusters, but no frame with such chains of greater length validates L. Syntactically, logics of depth n can be defined as those extensions of **K4** that contain the formula $\boldsymbol{bd}_n$ but not $\boldsymbol{bd}_{n+1}$, where

$$\begin{aligned} \boldsymbol{bd}_1 &= \Diamond\Box p_1 \to p_1, \\ \boldsymbol{bd}_{n+1} &= \Diamond(\Box p_{n+1} \wedge \neg \boldsymbol{bd}_n) \to p_{n+1}. \end{aligned}$$

The following theorem was proved in [129]:

THEOREM 24 (Segerberg). *Every logic of finite depth has the finite model property* (*in fact, is locally tabular in the sense that it has only finitely many nonequivalent formulas with variables $p_1, \dots, p_n$*), *and so is decidable if finitely axiomatisable.*

It is to be noted that there are a continuum of logics of depth 3 [84].

Say that a logic $L \in \mathrm{NExt}\,\mathbf{K4}$ is of *width* $n < \omega$ if it has a rooted frame $\langle W, R\rangle$ with an antichain $x_1, \dots, x_n$ (i.e., $x_i R x_j$ does not hold for any distinct $i, j \leq n$) but no rooted frame with an $n+1$-point antichain validates L. Syntactically, logics of width n can be described as those extensions of $\mathbf{K4}$ that contain the formula $\boldsymbol{bw}_n$ but not $\boldsymbol{bw}_{n+1}$, where

$$\boldsymbol{bw}_n \quad = \quad \bigwedge_{i=0}^{n} \Diamond p_i \to \bigvee_{0 \leq i \neq j \leq n} \Diamond\big(p_i \wedge (p_j \vee \Diamond p_j)\big).$$

The logics of width 1 are precisely the extensions of $\mathrm{NExt}\,\mathbf{K4.3}$.

The following theorem was proved in [43]:

THEOREM 25 (Fine). *All logics of finite width are Kripke complete.*

There are a continuum of logics of width 1. However, those of them that are finitely axiomatisable behave quite nicely as was shown in [176, 93]:

THEOREM 26. *All finitely axiomatisable logics in* $\mathrm{NExt}\,\mathbf{K4.3}$ *are decidable* (*in fact coNP-complete*), *though not necessarily have the finite model property.*

Nothing is known about decidability of finitely axiomatisable logics of width $n > 1$ (our conjecture is that all of them are decidable):

PROBLEM 11. Are finitely axiomatisable logics of width $n > 1$ decidable? What is their computational complexity?

For logics above $\mathbf{S4.3}$ we have the following classical result of [14, 40, 139]. Here and in what follows we say that a logic L has the *poly-size model property* if every formula $\varphi \notin L$ is refuted in a model based of a frame for L of polynomial size.

THEOREM 27 (Bull, Fine, Spaan). *All logics in* $\mathrm{NExt}\,\mathbf{S4.3}$ *are finitely axiomatisable, have the poly-size model property, and are coNP-complete.*

PROBLEM 12. Does there exist an algorithm that decides, given a formula φ, whether the logic $\mathbf{K4} \oplus \varphi$ is of finite width/depth?

It is worth noting that if the problem whether $L = \mathbf{K4} \oplus \varphi$ is of depth depth (or, which is equivalent, whether L is locally tabular) is decidable, then the tabularity problem for $\mathbf{K4}$ (that is, Problem 10) is decidable as well. Indeed, suppose that we have an algorithm for deciding, given a formula φ, whether $\mathbf{K4} \oplus \varphi$ is locally tabular. If this hypothetical algorithm says that $L = \mathbf{K4} \oplus \varphi$ is not locally tabular then L is not tabular either. Otherwise, we can effectively find some number n such that $\boldsymbol{bd}_n \in L$. And then we use Blok's [10] result according to which there are only *finitely many* pretabular logics containing $\boldsymbol{bd}_n$. All these pretabular logics have rather simple Kripke frames which can be easily axiomatised, so all of them are decidable. What remains to be done is to run Kuznetsov's algorithm described above.

Logics containing K5

Recall that $\mathbf{K5} = \mathbf{K} \oplus \Diamond\Box p \to \Box p$ is the logic determined by all *Euclidean* frames $\langle W, R\rangle$, where Euclidean means that

$$\forall x \forall y \forall z (xRz \wedge xRy \to yRz).$$

The papers [113, 114] investigate the (possibly non-normal) extensions of **K5**. The following theorem summarises the results for logics in NExt **K5**:

THEOREM 28 (Nagle, Thomason). *All logics in* NExt **K5** *have the finite model property, are finite axiomatisable, and so decidable. The lattice* NExt **K5** *can be computed effectively.*

It is not difficult to see that actually all logics in NExt **K5** have the poly-size model property and are coNP-complete.

Logics containing $\mathbf{S5} \times \mathbf{S5}$

$\mathbf{S5} \times \mathbf{S5}$ is the bimodal logic determined by product frames of the form $\langle W \times W, R_1, R_2 \rangle$, where $(w_1, w_2)R_1(w_1', w_2')$ iff $w_2 = w_2'$, and $(w_1, w_2)R_2(w_1', w_2')$ iff $w_1 = w_1'$. It was introduced and investigated because of its close relation to the two-variable fragment of first-order logic [130, 48]; see also Chapter 15. $\mathbf{S5} \times \mathbf{S5}$ can be axiomatised by adding to the fusion (see Chapter 15) of **S5** with **S5** the modal axiom $\Diamond_1 \Diamond_2 p \leftrightarrow \Diamond_2 \Diamond_1 p$ saying that R_1 and R_2 commute. Logics containing $\mathbf{S5} \times \mathbf{S5}$ are surprisingly well-behaved. Indeed, while $\mathbf{S5} \times \mathbf{S5}$ itself is NEXPTIME-complete [103], it was proved in [6, 5] that we have the following:

THEOREM 29. *Every normal bimodal logic properly containing* $\mathbf{S5} \times \mathbf{S5}$ *has the poly-size model property, is finitely axiomatisable and coNP-complete.*

The axiomatisation result is based on a variant of Kruskal's tree theorem. The picture is different if a constant for the diagonal $\{(w, w) \mid w \in W\}$ (motivated by its interpretation as 'equality' in first-order logic) is added to the bimodal language. In this case there exist uncountably many normal modal logics extending $\mathbf{S5} \times \mathbf{S5}$ with the diagonal, and it is open whether all of them have the finite model property [3, 4].

Logics containing $\mathbf{K} \oplus \mathbf{alt}_n$

A frame $\mathfrak{F} = \langle W, R \rangle$ validates the formula

$$\boldsymbol{alt}_n \quad = \quad \Box p_1 \vee \Box(p_1 \to p_2) \vee \cdots \vee \Box(p_1 \wedge \cdots \wedge p_n \to p_{n+1}),$$

where $n \geq 0$, iff each point in $\mathfrak{F}$ has at most n distinct R-successors. Segerberg [131] proved the following:

THEOREM 30. *All logics in* NExt $(\mathbf{K} \oplus \boldsymbol{alt}_1)$ *have the finite model property, are finitely axiomatisable, and so decidable. The lattice* NExt $(\mathbf{K} \oplus \boldsymbol{alt}_1)$ *can be computed effectively.*

It is not difficult to see that actually all extensions of $\mathbf{K} \oplus \boldsymbol{alt}_1$ have the poly-size model property and are coNP-complete. Extensions of $\mathbf{K} \oplus \boldsymbol{alt}_n$ for $n > 1$ are investigated in [1]:

THEOREM 31. *All logics in* NExt $(\mathbf{K} \oplus \boldsymbol{alt}_n)$ *are Kripke complete and their frames are first-order definable.*

An analysis of polymodal extensions of $\mathbf{K} \oplus \boldsymbol{alt}_n$ is given in [76, 61].

6 FRAME-THEORETIC CHARACTERISATION

Finding characterisations of those classes of structures that can be defined by (sets of) formulas of a given language $\mathcal{L}$ is one of the central research problems in the development of a model theory for the language. This is often achieved by introducing certain truth-preserving operators on classes of structures (e.g., the formation of p-morphic images, generated subframes, disjoint unions, ultraproducts, etc.) and then proving that the $\mathcal{L}$-definable classes are precisely those that are closed under these operators—a kind of Birkhoff-type theorem for varieties of abstract algebras. Such characterisations for modal logics are discussed in Chapters 5 and 6 on modal model theory and algebras. Unfortunately, abstract characterisations of this sort are of limited use when we deal with modal decision problems. In this context, what we need is not characterisations that are 'as abstract as possible,' but rather *explicit finitely presentable* ones.

Of course, modal formulas themselves can be regarded as a 'finitely presented characterisation' of modally definable classes of frames. However, in general, the information contained in formulas is rather *implicit* and non-structural (or 'non-geometric')—one has to work hard to learn how to decipher their meaning.

As a first step towards more informative finite presentations of modally definable classes of general frames, let us find out which of these classes $\mathcal{F}$ cannot be *decomposed* in the sense that whenever $\mathcal{F} = \mathsf{GFr}\,\Gamma$, for some set Γ of modal formulas, then there is a $\psi \in \Gamma$ such that $\mathcal{F} = \mathsf{GFr}\,\{\psi\}$. This means, in particular, that we cannot make the information provided by such a formula ψ 'more explicit' by replacing it with two (or more) formulas ψ_1 and ψ_2 such that $\mathsf{GFr}\,\{\psi\} = \mathsf{GFr}\,\{\psi_1, \psi_2\}$, but $\mathsf{GFr}\,\{\psi\} \subsetneqq \mathsf{GFr}\,\{\psi_i\}$ for $i = 1, 2$.

Again, as in Blok's dichotomy and Chagrov's classification, it is the notion of splittings that provides a proper framework for investigating indecomposability. In fact, one can show that, for every normal modal logic L, the class $\mathsf{GFr}\,L$ is indecomposable iff there exists a finite rooted (cycle free) frame $\mathfrak{F}$ such that $L = \mathbf{K}/\mathfrak{F}$. Indeed, suppose that $L = \mathbf{K}/\mathfrak{F} = \mathbf{K} \oplus \Gamma$ for some set of formulas Γ. Then there is a $\psi \in \Gamma$ such that $\mathfrak{F} \not\models \psi$ (for otherwise $\mathfrak{F} \models \Gamma$ and therefore $\mathfrak{F} \in \mathsf{GFr}\,L$). But then $L \subseteq \mathbf{K} \oplus \psi$, from which (since $\psi \in \Gamma$) $L = \mathbf{K} \oplus \psi$. (For the other direction and further details see Chapter 8.) Thus, we can say that the formula ψ describes $\mathcal{F}$. Moreover, in view of (4), ψ is deductively equivalent to the formula $\Box^{\leq n}\delta_{\mathfrak{F}} \to \neg p_r$, where $n = d(\mathfrak{F})$, which explicitly says: '$\mathfrak{G} \in \mathcal{F}$ *iff there does not exist a generated subframe of* $\mathfrak{G}$ *having* $\mathfrak{F}$ *as its p-morphic image.*' The formula $\Box^{\leq n}\delta_{\mathfrak{F}} \to \neg p_r$ can be regarded as a *modal diagram* of $\mathfrak{F}$.

It follows from these considerations that we would have a kind of optimal explicit and finite presentation of modally definable classes if we could prove that for every formula φ there exists a set $\mathcal{F}$ of finite rooted frames such that

$$\mathsf{GFr}\,\varphi \quad = \quad \mathsf{GFr}\,(\mathbf{K}/\mathcal{F}).$$

Then every modally definable class could be presented by means of a set of indecomposable geometrically explicit formulas as above.

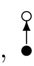
Now, the *bad news* is that we know from the proof of Blok's dichotomy that this is far from being the case: $\mathbf{D}$ is the only standard modal system that can be represented in this way.

But the *good news* is that this situation changes drastically as soon as we confine ourselves to 'transitive' modal logics, in particular, unimodal normal extensions of $\mathbf{K4}$,

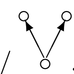
linear tense logics, or bimodal provability logics. Although, as we shall see below, it is impossible to represent all normal modal logics extending, say, **K4** in the form $L = \mathbf{K4}/\mathcal{F}$, for some set $\mathcal{F}$ of finite rooted frames, we can still introduce appropriate modifications and extensions of the notion of splitting which allow geometric finite representations of every normal extension of **K4**. In this chapter we will show such representations first for normal extensions of **K4** and then for extensions of linear tense logics. The case of bimodal provability logics is considered in [167].

6.1 Canonical formulas for **K4**

This frame-theoretic or 'geometric' approach to investigating modal logics in NExt **K4** and similar classes was launched by Jankov [66, 69] (in the framework of extensions of intuitionistic logic[6]), Blok [9], Fine [45] and Zakharyaschev [172, 174, 175]. Let us observe first that a number of standard logics in NExt **K4** are indeed union-splittings, and so their frames can be elegantly characterised in frame-theoretic terms. For example,

$\mathbf{S4} = \mathbf{K4} \oplus \Box p \to p = \mathbf{K4}/\{\bullet,$  $\}$,

$\mathbf{S4.1} = \mathbf{S4} \oplus \Box\Diamond p \to \Diamond\Box p = \mathbf{S4}/②$,

$\mathbf{S4.2} = \mathbf{S4} \oplus \Diamond\Box p \to \Box\Diamond p = \mathbf{S4}/$  ,

where ② is a two-point cluster. As we saw above, this means that, e.g., for every (general) frame $\mathfrak{F}$ for **S4**, $\mathfrak{F} \not\models \Box\Diamond p \to \Diamond\Box p$ iff there is a generated subframe of $\mathfrak{F}$ which can be p-morphically mapped onto ②. To appreciate the elegance of this frame-theoretic language, compare the purely geometric characterisation above with the standard first-order description of the Kripke frames for **S4.1**:

$$\langle W, R\rangle \models \Box\Diamond p \to \Diamond\Box p \qquad \text{iff} \qquad \forall x \exists y \; \big(xRy \wedge \forall z \; (yRz \to y = z)\big).$$

This observation leads to the following natural questions:

(A) Is it possible to characterise transitive frames for arbitrary formulas in a similar way?

(B) If this is indeed the case, then perhaps the decision problem (as well as many other problems) could be reduced to 'comparing' some finite frames? (For example, $\mathbf{K4}/\mathfrak{F} \subseteq \mathbf{K4}/\mathfrak{G}$ iff $\mathfrak{G}$ is a p-morphic image of some generated subframe of $\mathfrak{F}$.)

We analyse these questions using a number of simple examples. Consider first the Gödel–Löb provability logic $\mathbf{GL} = \mathbf{K4} \oplus \boldsymbol{la}$, where

$$\boldsymbol{la} \quad = \quad \Box(\Box p \to p) \to \Box p.$$

It is well-known that a Kripke frame $\mathfrak{F}$ validates $\boldsymbol{la}$ iff $\mathfrak{F}$ is transitive, irreflexive (i.e., a strict partial order) and *Noetherian* in the sense that it contains no infinite ascending

[6] Jankov [69] described all 'conjunctively indecomposable' intuitionistic formulas—i.e., splittings of the extensions of intuitionistic logic—and promised to investigate 'decomposable formulas' in his next paper which has never appeared. At the beginning of the 1980s he was arrested by the KGB for his support of the Solidarity movement in Poland.

chain. It is also well-known that the condition of Noetherianness is not a first-order one. But what is more important in the present context, the frame

$$\circ \quad \cdots \rightarrow\bullet\rightarrow\bullet\rightarrow\bullet$$

refutes $\boldsymbol{la}$ and yet contains no generated subframe that can be p-morphically mapped onto a finite frame refuting $\boldsymbol{la}$. This means that **GL** is not a union-splitting of NExt **K4** by means of finite frames. To find an explicit finitely presentable geometric characterisation of frames for **GL** some other frame-theoretic constructions are needed. Let us have another look at the structure of countermodels for $\boldsymbol{la}$.

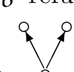
Suppose that a general frame $\mathfrak{F} = \langle W, R, P\rangle$ refutes $\boldsymbol{la}$ under some valuation. Then the set $V = \{x \in W \mid x \not\models \boldsymbol{la}\}$ is in P and $V \subseteq V{\downarrow} = \{w \in W \mid \exists v \in V\ wRv\}$. It follows from the former that $\mathfrak{G} = \langle V, R{\restriction}V, \{X \cap V \mid X \in P\}\rangle$ is a frame—we call it the *subframe of* $\mathfrak{F}$ *induced by* V. And the latter condition means that there is a p-morphism from $\mathfrak{G}$ onto a single reflexive point $\circ$, which is the simplest refutation frame for $\boldsymbol{la}$. Moreover, one can readily check that the converse also holds: if there is a subframe $\mathfrak{G}$ of $\mathfrak{F}$ which can be p-morphcally mapped onto $\circ$ then $\mathfrak{F} \not\models \boldsymbol{la}$.

This example motivates the following definitions. Given frames $\mathfrak{F} = \langle W, R, P\rangle$ and $\mathfrak{G} = \langle V, S, Q\rangle$, a partial (i.e., not totally defined, in general) map f from W onto V is called a *subreduction* of $\mathfrak{F}$ to $\mathfrak{G}$ if, for all $x, y \in \operatorname{dom} f = f^{-1}(V)$ and all $X \in Q$, it satisfies the following conditions

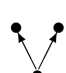
- xRy implies $f(x)Sf(y)$;
- $f(x)Sf(y)$ implies $\exists z \in W\ \big(xRz \wedge f(z) = f(y)\big)$;
- $f^{-1}(X) \in P$.

In other words, an f-*subreduct* of $\mathfrak{F}$ is a p-morphic image—or a *reduct*—of the subframe of $\mathfrak{F}$ induced by $\operatorname{dom} f$. A frame $\mathfrak{G} = \langle V, S, Q\rangle$ is a *subframe* of $\mathfrak{F} = \langle W, R, P\rangle$ if $V \subseteq W$ and the identity map on V is a subreduction of $\mathfrak{F}$ to $\mathfrak{G}$, i.e., if $S = R{\restriction}V$ and $Q \subseteq P$. Note that a generated subframe $\mathfrak{G}$ of $\mathfrak{F}$ is not in general a subframe of $\mathfrak{F}$, since V may be not in P.

Thus, the characterisation of frames for **GL** can be reformulated like this: $\mathfrak{F} \not\models \boldsymbol{la}$ iff $\mathfrak{F}$ is subreducible to $\circ$. Here are two more examples:

- A frame $\mathfrak{F}$ refutes the *Grzegorczyk axiom* $\Box(\Box(p \to \Box p) \to p) \to p$ iff it is subreducible to $\bullet$ or to ②.
- A quasi-order $\mathfrak{F}$ refutes the *Dummett axiom* $\Box(\Box p \to q) \vee \Box(\Box q \to p)$ iff $\mathfrak{F}$ is subreducible to  .

Now let us consider the logic $\mathbf{GL.2} = \mathbf{GL} \oplus \boldsymbol{ga}$, where $\boldsymbol{ga}$ is the *Geach axiom* $\Diamond\Box p \to \Box\Diamond p$. It is easy to see that every Kripke frame refuting $\boldsymbol{ga}$ must contain the fork



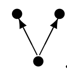
as a subframe, and in general, if $\mathfrak{F} \not\models \boldsymbol{ga}$ then $\mathfrak{F}$ is subreducible to this fork. However, the converse does not hold—just add a point $\bullet$ above the spikes of the fork to obtain a

counterexample. What we actually need is a fork-like subframe with its spikes having no common successor. A good mathematical notion that is capable of describing this and other similar cases is the notion of cofinality.

A subreduction f of $\mathfrak{F}$ to $\mathfrak{G}$ is called *cofinal* if

$$\mathrm{dom} f\uparrow \quad \subseteq \quad \mathrm{dom}\, f \cup \mathrm{dom}\, f\downarrow$$

or in English: if a point x is accessible from the domain of f then either x belongs to the domain of f itself or 'sees' a point in $\mathrm{dom}\, f$. For example, if we add to the fork a top point as above, then the resulting frame is subreducible to the original fork, but *not cofinally* because the top point cannot belong to the domain of the subreduction.

Returning back to the Geach axiom $\boldsymbol{ga}$, it is an easy exercise to show that a frame $\mathfrak{F}$ for **GL** refutes $\boldsymbol{ga}$ iff $\mathfrak{F}$ is cofinally subreducible to . Another example: a transitive $\mathfrak{F}$ refutes $\Diamond\top$ iff $\mathfrak{F}$ is cofinally subreducible to $\bullet$.

For the majority of standard modal axioms these two notions—plain and cofinal subreductions—are enough. But not for all. The simplest counterexample is the *density axiom* $\boldsymbol{den} = \Box\Box p \to \Box p$. It is refuted by the chain $\mathfrak{H}$ of two irreflexive points but becomes valid if we insert between them a reflexive one. In fact, $\mathfrak{F} \not\models \boldsymbol{den}$ iff there is a subreduction f of $\mathfrak{F}$ to $\mathfrak{H}$ such that $f(x\uparrow) = \{a\}$ for no point x in $\mathrm{dom}\, f\uparrow - \mathrm{dom}\, f$, where a is the final point in $\mathfrak{H}$.

Intuitively, every refutation frame for formulas like $\boldsymbol{la}$ can be constructed by adding new points to a frame $\mathfrak{G}$ that is reducible to some finite refutation frame of fixed size. For formulas like $\boldsymbol{ga}$ we have to take into account the cofinality condition and do not place new points 'above' $\mathfrak{G}$. And formulas like $\boldsymbol{den}$ impose another restriction: some places inside $\mathfrak{G}$ may be 'closed' for inserting new points. These 'closed domains' can be singled out in the following way.

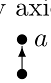
Suppose $\mathfrak{N} = \langle \mathfrak{H}, \mathfrak{U} \rangle$ is a model and $\mathfrak{a}$ an *antichain* in $\mathfrak{H}$—i.e., the points in $\mathfrak{a}$ do not see each other. Say that $\mathfrak{a}$ is an *open domain* in $\mathfrak{N}$ relative to a formula φ if there is a pair $t_{\mathfrak{a}} = (\Gamma_{\mathfrak{a}}, \Delta_{\mathfrak{a}})$ such that $\Gamma_{\mathfrak{a}} \cup \Delta_{\mathfrak{a}} = \mathsf{sub}\,\varphi$, $\bigwedge \Gamma_{\mathfrak{a}} \to \bigvee \Delta_{\mathfrak{a}} \notin \mathbf{K4}$ and

- $\Box\psi \in \Gamma_{\mathfrak{a}}$ implies $\psi \in \Gamma_{\mathfrak{a}}$,
- $\Box\psi \in \Gamma_{\mathfrak{a}}$ iff $a \models \Box^+\psi$ for all $a \in \mathfrak{a}$.

Otherwise $\mathfrak{a}$ is called a *closed domain* in $\mathfrak{N}$ relative to φ. A reflexive singleton $\mathfrak{a} = \{a\}$ is always open: just take $t_{\mathfrak{a}} = (\{\psi \in \mathsf{sub}\,\varphi \mid a \models \psi\}, \{\psi \in \mathsf{sub}\,\varphi \mid a \not\models \psi\})$. It is easy to see also that antichains consisting of points from the same clusters are open or closed simultaneously; we will not distinguish between such antichains.

Given a frame $\mathfrak{H}$ and a (possibly empty) set $\mathfrak{D}$ of antichains in $\mathfrak{H}$, we say that a subreduction f of $\mathfrak{F}$ to $\mathfrak{H}$ satisfies the *closed domain condition* for $\mathfrak{D}$ if

$$\text{(CDC)} \qquad \neg\exists x \in \mathrm{dom}\, f\uparrow - \mathrm{dom}\, f \;\exists \mathfrak{d} \in \mathfrak{D} \quad f(x\uparrow) \;=\; \mathfrak{d} \cup \mathfrak{d}\uparrow.$$

In terms of (CDC) refutation frames for the density axiom $\boldsymbol{den}$ can be characterised as follows: $\mathfrak{F} \not\models \boldsymbol{den}$ iff there is a subreduction of $\mathfrak{F}$ to  satisfying (CDC) for $\{\{a\}\}$.

Suppose now that $\mathfrak{N} = \langle \mathfrak{H}, \mathfrak{U} \rangle$ is a finite countermodel for φ and $\mathfrak{D}$ is the set of all closed domains in $\mathfrak{N}$ relative to φ. We claim that in this case $\mathfrak{F} \not\models \varphi$ whenever there is a cofinal subreduction f of $\mathfrak{F}$ to $\mathfrak{H}$ satisfying (CDC) for $\mathfrak{D}$. Moreover, if φ is *negation free* (i.e., contains no $\bot$, $\neg$, $\Diamond$) then a plain subreduction satisfying (CDC) for $\mathfrak{D}$ is enough. Indeed, if f is cofinal and $\mathfrak{F} = \langle W, R, P \rangle$ then we can assume that $\operatorname{dom} f \cup \operatorname{dom} \uparrow = W$. Define a valuation $\mathfrak{V}$ in $\mathfrak{F}$ as follows. If $x \in \operatorname{dom} f$ then we take $x \models p$ iff $f(x) \models p$, for every variable p in φ. If $x \notin \operatorname{dom} f$ then $f(x\uparrow) \neq \emptyset$, since f is cofinal. Let $\mathfrak{a}$ be an antichain in $\mathfrak{H}$ such that $\mathfrak{a} \cup \mathfrak{a} \uparrow = f(x\uparrow)$. By (CDC), $\mathfrak{a}$ is an open domain in $\mathfrak{N}$, and we put $y \models p$ iff $p \in \Gamma_{\mathfrak{a}}$, for every $y \notin \operatorname{dom} f$ such that $f(y\uparrow) = f(x\uparrow)$. It is easy to check that under this valuation $x \models \psi$ iff $f(x) \models \psi$ in the case $x \in \operatorname{dom} f$, and $x \models \psi$ iff $\psi \in \Gamma_{\mathfrak{a}}$, where $\mathfrak{a}$ is the open domain in $\mathfrak{N}$ associated with x, in the case $x \notin \operatorname{dom} f$, for every $\psi \in \mathsf{sub}\, \varphi$. If φ is negation free and f is a plain subreduction then $f(x\uparrow)$ may be empty. In such a case we just put $x \models p$, for all variables p.

Moreover, given an arbitrary formula φ, one can effectively construct a finite collection of finite rooted frames $\mathfrak{F}_1, \dots, \mathfrak{F}_n$ (of some fixed size that depends on the size of φ) and select in them sets $\mathfrak{D}_1, \dots, \mathfrak{D}_n$ of antichains such that, for any frame $\mathfrak{F}$, $\mathfrak{F} \not\models \varphi$ iff there is a cofinal subreduction of $\mathfrak{F}$ to $\mathfrak{F}_i$, for some i, satisfying (CDC) for $\mathfrak{D}_i$. If φ is negation free then a plain subreduction satisfying (CDC) is enough. Details can be found in [172, 24].

This 'explicit finitely presentable' characterisation of the constitution of refutation transitive frames can be expressed in the language of modal formulas similarly to the equation

$$\mathbf{K}/\mathfrak{F} \quad = \quad \mathbf{K} \;\oplus\; \Box^{\leq n}\delta_{\mathfrak{F}} \to \neg p_r.$$

Indeed, with every finite frame $\mathfrak{F} = \langle W, R \rangle$ with root r and every (possibly empty) set $\mathfrak{D}$ of antichains in $\mathfrak{F}$ we can associate formulas $\alpha(\mathfrak{F}, \mathfrak{D}, \bot)$ and $\alpha(\mathfrak{F}, \mathfrak{D})$ such that $\mathfrak{G} \not\models \alpha(\mathfrak{F}, \mathfrak{D}, \bot)$ ($\mathfrak{G} \not\models \alpha(\mathfrak{F}, \mathfrak{D})$) iff there is a cofinal (respectively, plain) subreduction of $\mathfrak{G}$ to $\mathfrak{F}$ satisfying (CDC) for $\mathfrak{D}$. Consider, for example, the following formulas

$$\alpha(\mathfrak{F}, \mathfrak{D}, \bot) \quad = \quad \Box^{+}\delta(\mathfrak{F}, \mathfrak{D}) \to \neg p_r,$$

where

$$\begin{aligned}
\delta(\mathfrak{F}, \mathfrak{D}) \quad = \quad & \bigwedge_{xRy} (p_x \to \Diamond p_y) \wedge \bigwedge_{\neg xRy} (p_x \to \neg\Diamond p_y) \wedge \bigwedge_{x \neq y} (p_x \to \neg p_y) \wedge \\
& \bigwedge_{\mathfrak{d} \in \mathfrak{D}} \Big(\bigwedge_{\substack{x \in \mathfrak{d} \cup \mathfrak{d}\uparrow \\ y \notin \mathfrak{d} \cup \mathfrak{d}\uparrow}} (\Diamond p_x \wedge \neg\Diamond p_y) \to \bigvee_{z \in W} p_z \Big) \wedge \\
& \bigwedge_{x \in W} (p_x \to \Diamond \bigvee_{y \in W} p_y)
\end{aligned}$$

and $\alpha(\mathfrak{F}, \mathfrak{D})$ is obtained by omitting the last conjunct from $\delta(\mathfrak{F}, \mathfrak{D})$. The formulas $\alpha(\mathfrak{F}, \mathfrak{D}, \bot)$ and $\alpha(\mathfrak{F}, \mathfrak{D})$ (or any other deductively equivalent formulas) are called the *canonical* and *negation free canonical formulas* for $\mathfrak{F}$ and $\mathfrak{D}$, respectively (it is not hard to get rid of $\neg$ and $\Diamond$ in the latter formula; see, e.g., [24]). The semantical meaning of these formulas should be clear: $\alpha(\mathfrak{F}, \mathfrak{D}, \bot)$ is refuted in a frame $\mathfrak{G}$ iff there is a cofinal subreduction of $\mathfrak{G}$ to $\mathfrak{F}$ satisfying (CDC) for $\mathfrak{D}$.

D4	$=$	$\mathbf{K4} \oplus \alpha(\bullet, \bot)$
S4	$=$	$\mathbf{K4} \oplus \alpha(\bullet)$
GL	$=$	$\mathbf{K4} \oplus \alpha(\circ)$
Grz	$=$	$\mathbf{K4} \oplus \alpha(\bullet) \oplus \alpha(②)$
K4.1	$=$	$\mathbf{K4} \oplus \alpha(\bullet, \bot) \oplus \alpha(②, \bot)$
Triv	$=$	$\mathbf{K4} \oplus \alpha(\bullet) \oplus \alpha(②) \oplus \alpha\left(\begin{smallmatrix}\circ\\ \uparrow\\ \circ\end{smallmatrix}\right)$
Verum	$=$	$\mathbf{K4} \oplus \alpha(\circ) \oplus \alpha\left(\begin{smallmatrix}\bullet\\ \uparrow\\ \bullet\end{smallmatrix}\right)$
S5	$=$	$\mathbf{S4} \oplus \alpha\left(\begin{smallmatrix}\circ\\ \uparrow\\ \circ\end{smallmatrix}\right)$
K4B	$=$	$\mathbf{K4} \oplus \alpha\left(\begin{smallmatrix}*\\ \uparrow\\ *\end{smallmatrix}\right)$ (4 axioms)
K4.2	$=$	$\mathbf{K4} \oplus \alpha\left(\begin{smallmatrix}\bullet\\ \uparrow\\ \bullet\\ \uparrow\\ \bullet\end{smallmatrix}, \bot\right) \oplus \alpha\left(\begin{smallmatrix}\bullet\\ \uparrow\\ \circ\end{smallmatrix}, \bot\right) \oplus \alpha\left(\begin{smallmatrix}* & & *\\ \nwarrow & & \nearrow\\ & * & \end{smallmatrix}, \bot\right)$ (8 axioms)
K4.3	$=$	$\mathbf{K4} \oplus \alpha\left(\begin{smallmatrix}* & & *\\ \nwarrow & & \nearrow\\ & * & \end{smallmatrix}\right)$ (6 axioms)
Dum	$=$	$\mathbf{S4} \oplus \alpha\left(\begin{smallmatrix}② & & \circ\\ \nwarrow & & \nearrow\\ & \circ & \end{smallmatrix}\right) \oplus \alpha\left(\begin{smallmatrix}\circ\\ \uparrow\\ ②\end{smallmatrix}\right)$

Table 1. Canonical axioms of standard modal logics

THEOREM 32. *There is an algorithm which, given an $\mathcal{ML}_1$-formula φ, returns canonical formulas $\alpha(\mathfrak{F}_1, \mathfrak{D}_1, \bot), \dots, \alpha(\mathfrak{F}_n, \mathfrak{D}_n, \bot)$ such that*

$$\mathbf{K4} \oplus \varphi = \mathbf{K4} \oplus \alpha(\mathfrak{F}_1, \mathfrak{D}_1, \bot) \oplus \cdots \oplus \alpha(\mathfrak{F}_n, \mathfrak{D}_n, \bot).$$

If φ is negation free then one can use negation free canonical formulas.

Table 1 shows canonical axiomatisations of some standard modal logics in the field of **K4**. For brevity we write $\alpha(\mathfrak{F}, \bot)$ instead of $\alpha(\mathfrak{F}, \emptyset, \bot)$ and $\alpha(\mathfrak{F})$ instead of $\alpha(\mathfrak{F}, \emptyset)$. Each $*$ in the table is to be replaced by both $\circ$ and $\bullet$.

Theorem 32 provides a solution to problem (A) formulated at the beginning of this section. It shows that as far as such properties of modal logics (from NExt **K4**) as decidability, completeness, the fmp, etc. are concerned, we can always deal with canonical formulas—which explicitly describe their frames. And the following observation gives a *partial* solution to problem (B).

THEOREM 33. (1) *For every logic $L = \mathbf{K4} \oplus \{\alpha(\mathfrak{F}_i, \emptyset) \mid i \in I\}$ and every canonical formula $\alpha(\mathfrak{F}, \mathfrak{D}, \bot)$, we have $\alpha(\mathfrak{F}, \mathfrak{D}, \bot) \in L$ iff $\mathfrak{F}$ is subreducible to $\mathfrak{F}_i$ for some $i \in I$.*

(2) *For every logic $L = \mathbf{K4} \oplus \{\alpha(\mathfrak{F}_i, \emptyset, \bot) \mid i \in I\}$ and every canonical formula $\alpha(\mathfrak{F}, \mathfrak{D}, \bot)$, we have $\alpha(\mathfrak{F}, \mathfrak{D}, \bot) \in L$ iff $\mathfrak{F}$ is cofinally subreducible to $\mathfrak{F}_i$ for some $i \in I$.*

(3) *For every logic $L = \mathbf{K4} \oplus \{\alpha(\mathfrak{F}_i, \mathfrak{D}_i, \bot) \mid i \in I\}$ and every canonical formula $\alpha(\mathfrak{F}, \mathfrak{D}^\sharp, \bot)$ where $\mathfrak{D}^\sharp$ is the set of all antichains in $\mathfrak{F}$, we have $\alpha(\mathfrak{F}, \mathfrak{D}^\sharp, \bot) \in L$ iff there is a generated subframe of $\mathfrak{F}$ that is reducible to $\mathfrak{F}_i$ for some $i \in I$.*

It follows from (3) that $\mathbf{K4} \oplus \varphi$ is a splitting of NExt $\mathbf{K4}$ iff φ is deductively equivalent in NExt $\mathbf{K4}$ to a formula of the form $\alpha(\mathfrak{F}, \mathfrak{D}^\sharp, \bot)$, where $\mathfrak{D}^\sharp$ is the set of all antichains in $\mathfrak{F}$ (in this case $\mathbf{K4}/\mathfrak{F} = \mathbf{K4} \oplus \alpha(\mathfrak{F}, \mathfrak{D}^\sharp, \bot)$). Such formulas are known as *Jankov formulas* (Jankov [66] introduced them for intuitionistic logic in the algebraic setting), or *frame formulas* (used by Fine [43]), or *Jankov–Fine formulas*.

But the most interesting consequence can be drawn from (1) or (2): all logics mentioned in (1) and (2) of Theorem 33 have the fmp, and so are decidable if finitely axiomatisable. It follows, for instance, that all logics in NExt $\mathbf{S4.3}$—which can be represented in the form (2) because antichains in frames for $\mathbf{S4.3}$ are reflexive singletons—have the fmp. It is not hard to see also that all these logics are finitely axiomatisable which gives the well-known result of Bull [14] and Fine [40] from Theorem 27.

As we have already mentioned, practically all 'standard' modal logics in the field of $\mathbf{K4}$ can be axiomatised by canonical formulas of the form $\alpha(\mathfrak{F}, \emptyset, \bot)$ or $\alpha(\mathfrak{F}, \emptyset)$. This yields an answer to the question 'why modal logic is so robustly decidable?' of Vardi [158] for the case of transitive unimodal logics. Although it is impossible to effectively recognise whether a logic $\mathbf{K4} \oplus \varphi$ can be axiomatised by such formulas [25], there is a simple model-theoretic characterisation of logics from (1) and (2) of Theorem 33 discovered in [45, 174]:

THEOREM 34. (1) *A logic $L \in$ NExt $\mathbf{K4}$ is axiomatisable by canonical formulas of the form $\alpha(\mathfrak{F}, \emptyset)$ iff L is characterised by a class of (general) frames that is closed under the formation of subframes.*

(2) *A logic $L \in$ NExt $\mathbf{K4}$ is axiomatisable by canonical formulas of the form $\alpha(\mathfrak{F}, \emptyset, \bot)$ iff L is characterised by a class of (general) frames that is closed under the formation of cofinal subframes.*

The logics from (1) and (2) are called *subframe* and *cofinal subframe logics*, respectively. It turns out that for these logics the notions of first-order definability, canonicity and strong Kripke completeness are equivalent; see [45, 174] and Theorem 44 below. It is worth noting that the fmp of (cofinal) subframe logics and the decidability of those of them that are finitely axiomatisable (there are a continuum of subframe logics [174]) is obtained from Theorems 32 and 33 for free: it suffices to check whether the frame of the tested canonical formula is (cofinally) subreducible to the frame of one of the canonical axioms of a given logic. This provides us with another general method of proving that a given logic is Kripke complete, decidable, canonical, has the finite model property, etc.: usually it is much easier to check that the class of general frames for a given logic is closed under cofinal subframes and to find the logic's canonical axioms than to use filtration and/or canonical models.

PROBLEM 13. Characterise the computational complexity of finitely axiomatisable cofinal subframe logics (e.g., all such extensions of $\mathbf{K4.3}$ are coNP-complete).

PROBLEM 14. Give a syntactical characterisation of subframe and cofinal subframe logics (cf. Theorem 53).

Note that, for every (cofinal) subframe logic L and every formula $\varphi \notin L$, there is a frame for L refuting φ whose size is exponential in the length of φ.

In general, question (B) does not seem to have such an elegant answer as for the case of cofinal subframe logics (see, however, [147] for a recent attempt to introduce 'canonical formulas' for NExt $\mathbf{K}$). We can only console ourselves with a number of sufficient

conditions of decidability, the fmp (in particular, Theorems 16 and 19) and some other properties imposed on the canonical axioms; for details consult [25, 175, 24, 177].

6.2 Canonical formulas for tense logics of linear time flows

In this section we consider normal bimodal logics containing the tense logic **Lin** of linear orders (the class of all frames $\langle W, R_1, R_2\rangle$ in which R_2 is the converse of R_1 and R_1 is a linear order, i.e., R_1 is transitive and connected $(\forall x \forall y xR_1y \vee yR_1x \vee x = y)$). These logics are of interest in this chapter for two reasons. First, many natural and useful modal logics are contained in this class, e.g., the logics determined by the flows of time $\langle \mathbb{N}, <\rangle$, $\langle \mathbb{Z}, <\rangle$, $\langle \mathbb{Q}, <\rangle$, and $\langle \mathbb{R}, <\rangle$ [15, 128, 135]. And second, this class is of exceptional interest because currently NExt **Lin** is the only lattice of modal logics for which *almost every decision problem is decidable* and which, nevertheless, contains numerous Kripke incomplete modal logics—where the standard techniques based on proving the finite model property or tree model property do not work.

The decidability results for logics in NExt **Lin** are based on two ingredients. First, they can be axiomatised by canonical formulas which explicitly show the geometrical and topological conditions they define. And second, a general completeness result establishes that they are determined by general frames which are composed from a set of rather simple 'atomic' general frames. In this part we introduce the canonical formulas. In the next section we discuss the general completeness theorem and survey its consequences regarding decision problems for logics in NExt **Lin**.

The logic **Lin** is obviously axiomatised as

$$\mathbf{Lin} \quad = \quad \mathbf{K4}_2 \oplus p \to \Box_1\Diamond_2 p \oplus p \to \Box_2\Diamond_1 p \oplus \Diamond_1\Diamond_2 p \vee \Diamond_2\Diamond_1 p \to p \vee \Diamond_1 p \vee \Diamond_2 p.$$

To begin our discussion of canonical formulas for logics extending NExt **Lin** observe that Dedekind cuts can be characterised by means of splittings with the frame $\circ\!\!\to\!\!\circ$. In fact, one can show that

$$\mathsf{Log}\,\{\langle \mathbb{R}, <, >\rangle\} \quad = \quad \mathsf{Log}\,\{\langle \mathbb{Q}, <, >\rangle\}/\ \circ\!\!\to\!\!\circ.$$

The intuition behind this equation should be clear from the observation that the class of rooted general frames for the logic $\mathsf{Log}\,\{\langle \mathbb{Q}, <, >\rangle\}/\ \circ\!\!\to\!\!\circ$ consists of all general frames $\left\langle W, R, R^{-1}, P\right\rangle$ such that

- $\langle W, R\rangle$ is a dense linear order without endpoints (i.e., it satisfies the properties $\forall x\forall y(xRy \to \exists z(xRz \wedge zRy))$, $\forall x \exists y\, xRy$ and $\forall x\exists y\, yRx$) and
- there is no p-morphism from $\left\langle W, R, R^{-1}, P\right\rangle$ onto $\circ\!\!\to\!\!\circ$.

But such a p-morphism exists iff there is a partition $X, Y \in P$ of W such that

$$\forall x \in X \forall y \in Y\ xRy, \qquad \forall x \in X \exists y \in X\ xRy, \qquad \forall x \in Y \exists y \in Y\ yRx. \tag{10}$$

Observe that $\langle \mathbb{Q}, <, >\rangle \notin \mathsf{Fr}\,\mathsf{Log}\,\{\langle \mathbb{Q}, <, >\rangle\}/\ \circ\!\!\to\!\!\circ$ since, e.g., $X = \{x \in \mathbb{Q} \mid x < \sqrt{2}\}$ and $Y = \{y \in \mathbb{Q} \mid y > \sqrt{2}\}$ form such a partition.

To characterise the class of *all linear orders* without partitions satisfying (10), we have to weaken the splitting formula. For example, the frame $\langle \{x \geq 0 \mid x \in \mathbb{Q}\}, <, >\rangle$ (which obviously has a partition satisfying (10)) validates

$$\mathbf{Lin}/\circ\!\!\to\!\!\circ,$$

simply because it contains an irreflexive left endpoint and, therefore, cannot be p-morphically mapped onto $\circ\!\!\rightarrow\!\!\circ$. To obtain the class of all linear orders without partitions satisfying (10) the point a in $\overset{a}{\circ}\!\!\rightarrow\!\!\overset{b}{\circ}$ should be regarded as reflexive by the operator $\Box_1$ for the future but as an arbitrary linear order by the operator $\Box_2$ for the past. To make this idea precise, we introduce the notion of a *type assignment* $\mathbf{t} = (\mathbf{t}_1, \mathbf{t}_2)$ which is a map from the set of clusters C of a finite linear order into the set of pairs over $\{\mathsf{j}, \mathsf{m}\}$ such that $\mathbf{t}C = (\mathbf{t}_1 C, \mathbf{t}_2 C) = (\mathsf{m}, \mathsf{m})$ for every cluster C consisting of an irreflexive point. Here j stands for 'joker' and m stands for 'maximal.' For example, $\mathbf{t}C = (\mathsf{m}, \mathsf{j})$ means intuitively that cluster C should be regarded as 'what it is' by $\Box_1$ and as an arbitrary linear order by $\Box_2$. The condition that irreflexive points are mapped to (m, m) means that they are always regarded as what they are, that is, irreflexive points. Before we associate with every finite linear order with type assignment $\langle \mathfrak{F}, \mathbf{t} \rangle$ a formula $\alpha(\mathfrak{F}, \mathbf{t})$ with the corresponding meaning some notation is required.

Given a finite sequence $\overline{\mathfrak{F}} = \left\langle \mathfrak{F}_i = \left\langle W_i, R_i, R_i^{-1}, P_i \right\rangle \mid 1 \le i \le n \right\rangle$ of disjoint frames, we denote by $[\overline{\mathfrak{F}}] = \mathfrak{F}_1 \lhd \cdots \lhd \mathfrak{F}_n$ the ordered sum of them, i.e., the frame $\left\langle W, R, R^{-1}, P \right\rangle$ in which

$$W \quad = \quad \bigcup_{i=1}^{n} W_i, \qquad R \quad = \quad \bigcup_{i=1}^{n} R_i \cup \bigcup_{1 \le i < j \le n} (W_i \times W_j)$$

and $P = \{X_1 \cup \cdots \cup X_n \mid X_i \in P_i\}$. Each finite frame can be represented then as the ordered sum $C_1 \lhd \cdots \lhd C_n$ of its clusters.

With every finite tense frame $\mathfrak{F} = \langle W, R \rangle = C_1 \lhd \cdots \lhd C_n$ with cluster assignment $\mathbf{t} = (\mathbf{t}_1, \mathbf{t}_2)$ we associate the formula

$$\alpha(\mathfrak{F}, \mathbf{t}) \quad = \quad \neg \overline{\delta}(\mathfrak{F}, \mathbf{t}),$$

where

$$\overline{\delta}(\mathfrak{F}, \mathbf{t}) \quad = \quad \delta(\mathfrak{F}, \mathbf{t}) \wedge \Box_2 \delta(\mathfrak{F}, \mathbf{t}) \wedge \Box_1 \delta(\mathfrak{F}, \mathbf{t})),$$

and

$$\begin{aligned}
\delta(\mathfrak{F}, \mathbf{t}) \quad = \quad & \bigwedge \{p_x \to \neg p_y \mid x \neq y\} \wedge \\
& \bigwedge \{p_x \to \neg \Diamond_1 p_y \mid \neg(xRy)\} \wedge \\
& \bigwedge \{p_x \to \Diamond_1 p_y \mid \exists i \le n \ (\mathbf{t}_1 C_i = \mathsf{m} \wedge x, y \in C_i \wedge xRy)\} \wedge \\
& \bigwedge \{p_x \to \Diamond_2 p_y \mid \exists i \le n \ (\mathbf{t}_2 C_i = \mathsf{m} \wedge x, y \in C_i \wedge xR^{-1}y)\} \wedge \\
& \bigvee \{p_y \mid y \in W\} \wedge \\
& \bigwedge \{\Diamond_1 p_y \vee p_y \vee \Diamond_2 p_y \mid y \in W\}.
\end{aligned}$$

To explain the semantical meaning of these formulas, notice first that if $\mathbf{t}C = (\mathsf{m}, \mathsf{m})$ for all clusters C then $\mathfrak{G} \not\models \alpha(\mathfrak{F}, \mathbf{t})$ is iff $\mathfrak{G}$ there exists a p-morphism from $\mathfrak{G}$ onto $\mathfrak{F}$. Therefore,

$$\mathbf{Lin}/\mathfrak{F} \quad = \quad \mathbf{Lin} \oplus \alpha(\mathfrak{F}, \mathbf{t}).$$

If $\mathbf{t}_i C = \mathsf{j}$ for some $i \in \{1, 2\}$ and some cluster C in $\mathfrak{F}$, then the formula $\alpha(\mathfrak{F}, \mathbf{t})$ can be refuted in frames that do not necessarily have $\mathfrak{F}$ as a p-morphic image. In this case $\mathfrak{G} \not\models \alpha(\mathfrak{F}, \mathbf{t})$ iff there exist frames $\mathfrak{G}_i$, for $1 \le i \le n$, such that $\mathfrak{G} = \mathfrak{G}_1 \lhd \cdots \lhd \mathfrak{G}_n$ and

$\mathbf{Ord}_t$	$=$	$\mathsf{Log}\,\{\langle\xi,<,>\rangle \mid \xi \text{ an ordinal}\} = \mathbf{Lin} \oplus \alpha(-,(\circ,(\mathsf{j},\mathsf{m})))$
$\mathbf{E}_t$	$=$	$\mathbf{Lin} \oplus \Diamond_1\top \oplus \Diamond_2\top =$
		$\mathbf{Lin} \oplus \alpha(-,(\bullet,(\mathsf{m},\mathsf{m}))) \oplus \alpha((\bullet,(\mathsf{m},\mathsf{m})),-)$
$\mathbf{RD}$	$=$	$\mathsf{Log}\,\{\mathfrak{G} \mid \forall x(\neg xRx \to \exists y(xRy \wedge \{z \mid xRzRy\} = \emptyset))\} =$
		$\mathbf{Lin} \oplus \alpha(-,(\bullet,(\mathsf{m},\mathsf{m}))) \oplus \alpha(-,(\bullet,(\mathsf{m},\mathsf{m})) \lhd (\circ,(\mathsf{m},\mathsf{j})))$
$\mathbf{LD}$	$=$	the mirror image of $\mathbf{RD}$
$\mathbf{Z}_t$	$=$	$\mathsf{Log}\,\langle\mathbb{Z},<,>\rangle =$
		$\mathbf{RD} \oplus \mathbf{LD} \oplus \alpha((\circ,(\mathsf{j},\mathsf{j})) \lhd (\circ,(\mathsf{j},\mathsf{m}))) \oplus \alpha((\circ,(\mathsf{m},\mathsf{j})) \lhd (\circ,(\mathsf{j},\mathsf{j})))$
$\mathbf{Ds}_n$	$=$	$\mathbf{Lin} \oplus \Box_1^{n+1}p \to \Box_1^n p =$
		$\mathbf{Lin} \oplus \alpha(-,\underbrace{(\bullet,(\mathsf{m},\mathsf{m}) \lhd \cdots \lhd (\bullet,(\mathsf{m},\mathsf{m}))}_{n+1},-)$
$\mathbf{Q}_t$	$=$	$\mathsf{Log}\,\langle\mathbb{Q},<,>\rangle = \mathbf{Ds}_1 \oplus \mathbf{E}_t$
$\mathbf{R}_t$	$=$	$\mathsf{Log}\,\langle\mathbb{R},<,>\rangle = \mathbf{Q}_t \oplus \alpha((\circ,(\mathsf{m},\mathsf{j})) \lhd (\circ,(\mathsf{j},\mathsf{m})))$

Table 2. Axiomatisations of standard tense logics

$\mathfrak{G}_i \not\models \alpha(C_i, \mathbf{t}\restriction C_i)$ for all $1 \leq i \leq n$. So it suffices to consider $\mathfrak{G} \not\models \alpha(C,\mathbf{t})$ for a cluster C. Assume for simplicity that $\mathfrak{G}$ is a Kripke frame. *Case 1:* $\mathbf{t}C = (\mathsf{j},\mathsf{j})$. Then $\mathfrak{G} \not\models \alpha(C,\mathbf{t})$ iff $|\mathfrak{G}| \geq |C|$. *Case 2:* $\mathbf{t}C = (\mathsf{m},\mathsf{j})$. Then C is nondegenerate and $\mathfrak{G} \not\models \alpha(C,\mathbf{t})$ iff either $\mathfrak{G}$ contains an R-final cluster of cardinality $\geq |C|$ or it has no R-final point at all. *Case 3:* $\mathbf{t}C = (\mathsf{j},\mathsf{m})$. This is the mirror image of Case 2. *Case 4:* $\mathbf{t}C = (\mathsf{m},\mathsf{m})$. If C is an irreflexive point then $\mathfrak{G}$ is an irreflexive point as well whenever $\mathfrak{G} \not\models \alpha(C,\mathbf{t})$. If C is non-degenerate and $\mathfrak{G} \not\models \alpha(C,\mathbf{t})$ then $\mathfrak{G}$ satisfies the conditions of Cases 2 and 3.

Now, the following is proved in [163]:

THEOREM 35. *There exists an algorithm which, given a formula φ, returns canonical formulas $\alpha(\mathfrak{F}_1,\mathbf{t}_1),\dots,\alpha(\mathfrak{F}_n,\mathbf{t}_n)$ such that*

$$\mathbf{Lin} \oplus \varphi \quad = \quad \mathbf{Lin} \ \oplus\ \alpha(\mathfrak{F}_1,\mathbf{t}_1) \ \oplus\ \dots\ \oplus\ \alpha(\mathfrak{F}_n,\mathbf{t}_n).$$

Canonical axiomatisations of some standard linear tense logics are shown in Table 2, where we use the following notation. Given a finite frame $\mathfrak{F} = C_1 \lhd \cdots \lhd C_n$, we write $\alpha((C_1,\mathbf{t}C_1) \lhd \cdots \lhd (C_n,\mathbf{t}C_n))$ instead of $\alpha(\mathfrak{F},\mathbf{t})$ and $\alpha(-,(C_1,\mathbf{t}C_1) \lhd \cdots \lhd (C_n,\mathbf{t}C_n))$ instead of

$$\alpha((C_1,\mathbf{t}C_1) \lhd \cdots \lhd (C_n,\mathbf{t}C_n)) \ \oplus\ \alpha((\circ,(\mathsf{j},\mathsf{j})) \lhd (C_1,\mathbf{t}C_1) \lhd \cdots \lhd (C_n,\mathbf{t}C_n)).$$

$\alpha((C_1,\mathbf{t}C_1) \lhd \cdots \lhd (C_n,\mathbf{t}C_n),-)$ is defined analogously.

Applications of this result will be discussed in the section below.

7 DECISION PROBLEMS FOR TENSE LOGICS

In this section we show that in NExt **Lin** almost all decision problems are decidable. For example, we can decide whether a finitely axiomatisable logic is Kripke complete, has the finite model property, is canonical, etc. Moreover, finitely axiomatisable linear tense logics turn out to be coNP-complete. Thus, NExt **Lin** is an example of a class of logics for which the original research programme has a complete and 'positive' solution.

The first step in understanding logics in NExt **Lin** was done in the section above, where we saw that they can be axiomatised by canonical formulas $\alpha(\mathfrak{F}, \mathbf{t})$ which show directly the geometric and topological conditions definable on linear orders by means of bimodal formulas. The second step is a completeness result which determines a class of rather simple general frames which respect to which every logic in NExt **Lin** is complete. We require the following notation:

(1) Denote by ⓚ the non-degenerate cluster with $k > 0$ points.

(2) Let $\omega^<(0)$ be the strictly ascending chain $\langle \omega, <, > \rangle$ of natural numbers, $\omega^<(1)$ the chain $\langle \omega, \leq, \geq \rangle$, $\omega^<(2)$ the ascending chain of natural numbers in which precisely the even points are reflexive, $\omega^<(3)$ the chain in which precisely the multiples of 3 are reflexive, and so on; $\omega^>(n)$ is the mirror image of $\omega^<(n)$.

(3) $\mathfrak{C}(0, ①) = \langle \omega^<(0) \lhd ①, P \rangle$, where P consists of all cofinite sets containing ① and their complements. We generalise this construction to chains $\omega^<(n)$ and clusters ⓚ. Namely, for $n < \omega$, $k > 1$ and ⓚ $= \{a_0, \ldots, a_{k-1}\}$, we put

$$\mathfrak{C}(n, ⓚ) \quad = \quad \langle \omega^<(n) \lhd ⓚ, P \rangle,$$

where P is the set generated by means of the Boolean operators from the set of finite subsets of $\omega^<(n)$ and the sets $\{X_i \mid 0 \leq i \leq k-1\}$, for $X_i = \{a_i\} \cup \{kj + i \mid j \in \omega\}$, $0 \leq i \leq k-1$. $\mathfrak{C}(ⓚ, n)$ denotes the mirror image of $\mathfrak{C}(n, ⓚ)$.

(4) $\mathfrak{C}(0, ①, 0) = \langle \omega^<(0) \lhd ① \lhd \omega^>(0), P \rangle$, where P consists of all cofinite sets containing ① and their complements.

It is easy to check that the frames defined in (3) and (4) are descriptive and a singleton $\{x\}$ is in P iff $x \notin$ ⓚ. Notice also that the logics $\mathsf{Log}(\mathfrak{C}(n, ⓚ))$, $\mathsf{Log}(\mathfrak{C}(ⓚ, n))$, $k \geq 2$, and $\mathsf{Log}\mathfrak{C}(0, ①, 0)$ are Kripke incomplete.

For a class of frames $\mathcal{C}$, we denote by $\mathcal{C}^*$ the class of finite sequences of frames from $\mathcal{C}$ and let $[\mathcal{C}^*] = \{[\overline{\mathfrak{F}}] \mid \overline{\mathfrak{F}} \in \mathcal{C}^*\}$. The class of finite clusters and the frames of the form (3) and (4) is denoted by $\mathcal{B}$. We are now in a position to formulate the completeness result:

THEOREM 36. *Each logic $L \in$ NExt **Lin** is determined by a set $\mathcal{C} \subseteq [\mathcal{B}^*]$.*

Proof. We briefly explain the idea of the proof. Suppose that $\mathfrak{M} = \langle \mathfrak{F}, \mathfrak{V} \rangle$ is a countermodel for $\alpha = \alpha((C_1, \mathbf{t}C_1) \lhd \cdots \lhd (C_n, \mathbf{t}C_n))$ based on a descriptive frame $\mathfrak{F} = \langle W, R, R^{-1}, P \rangle$. We must show that there exists $\mathfrak{G} \in [\mathcal{B}^*]$ refuting α and such that $\mathsf{Log}\,\mathfrak{G} \supseteq \mathsf{Log}\,\mathfrak{F}$. Consider the sets

$$W_i \quad = \quad \{y \in W \mid (\mathfrak{M}, y) \models \bigvee \{p_x \mid x \in C_i\}\}.$$

One can easily show that W_i are intervals in $\mathfrak{F}$ and $\mathfrak{F} = \mathfrak{F}_1 \lhd \cdots \lhd \mathfrak{F}_n$, for the subframes $\mathfrak{F}_i$ of $\mathfrak{F}$ induced by W_i. Moreover, $\mathfrak{G} = [\overline{\mathfrak{G}}]$ is as required if $\overline{\mathfrak{G}} = \langle \mathfrak{G}_1, \ldots, \mathfrak{G}_n \rangle$ is a sequence in $\mathcal{B}^*$ such that $\mathsf{Log}\,\mathfrak{G}_i \supseteq \mathsf{Log}\,\mathfrak{F}_i$, and $\mathfrak{G}_i \not\models \alpha(C_i, \mathbf{t}C_i)$, for $1 \leq i \leq n$. Frames $\mathfrak{G}_i$ with those properties are constructed in [163]. ❑

EXAMPLE 37. The logic $\mathbf{Q}_t$ of the rational numbers is determined by the frames $\mathfrak{F} \in [\mathcal{B}^*]$ which contain no pair of adjacent irreflexive points. The logic $\mathbf{R}_t$ of the real line is determined by the frames $\mathfrak{F} \in [\mathcal{B}^*]$ which contain neither a pair of adjacent irreflexive points nor a pair of adjacent non-degenerate clusters.

Now, based on both the canonical formulas for logics in NExt **Lin** and this completeness result, [93] shows:

THEOREM 38. (i) *All finitely axiomatisable logics in* NExt **Lin** *are coNP-complete.*
(ii) *All logics determined by a frame* $\mathfrak{G} \in [\mathcal{B}^*]$ *are coNP-complete.*

Proof. (ii) is proved by showing that, given a formula ψ, it can be checked in nondeterministic polynomial time in the length of ψ whether it is satisfiable in a given $\mathfrak{G} \in [\mathcal{B}^*]$. To this end, [93] shows that it is sufficient to check satisfiability of ψ in a certain finite subframe of $\mathfrak{G}$ (whose size is polynomial in ψ) under certain 'good' valuations.

(i) Suppose

$$L \quad = \quad \mathbf{Lin} \oplus \alpha(\mathfrak{F}_1, \mathbf{t}_1) \oplus \cdots \oplus \alpha(\mathfrak{F}, \mathbf{t}_n)$$

is given. Then [93] shows that any given formula ψ which is satisfiable in a frame $\mathfrak{G} \in [\mathcal{B}^*]$ validating L is satisfiable in a frame of this type whose parameters (i.e., the number of blocks required, the size of its clusters, and the maximal n such that $\mathfrak{C}(n, \textcircled{k})$ or $\mathfrak{C}(\textcircled{k}, n)$ occurs in it) are polynomial in ψ. By the proof of (ii), it can be checked in nondeterministic polynomial time whether ψ is satisfiable in such a frame. Additionally, one can show that it can be checked in polynomial time (in the length of ψ) whether such a frame validates a formula of the form $\alpha(\mathfrak{F}, \mathbf{t})$. ❑

Given the coNP-completeness of all finitely axiomatisable tense logics, at least two questions arise: First, are there interesting classes of finitely axiomatisable linear tense logics? Second, how complex are non-finitely axiomatisable linear tense logics? Regarding the second question, consider, for any $M \subseteq \mathbb{N}$, the logic L_M determined by the class of frames

$$\{\langle\{1, \ldots, m\}, <\rangle \mid m \in M\}.$$

Set $\bot = p \wedge \neg p$ and $\top = p \vee \neg p$. Then the formula

$$\varphi_m \quad = \quad \Box_2 \bot \wedge \Box_1^{m+1} \bot \wedge \Diamond_1^m \top$$

is satisfiable in a frame validating L_M iff $m+1 \in M$. Thus, for any set of natural numbers M there exists a tense logic of the same complexity as M. Regarding the first question, [163] determines a number of classes of finitely axiomatisable linear tense logics. For example, the following result is proved using Kruskal's tree theorem [80]:

THEOREM 39. *A linear tense logic L is finitely axiomatisable whenever there exists $n < \omega$ such that $\Box_1^{n+1} p \to \Box_1^n p \in L$. In particular, all linear tense logics of reflexive frames as well as all extensions of the tense logic of $\langle \mathbb{Q}, < \rangle$ are finitely axiomatisable.*

Where are the Kripke incomplete modal logics in NExt **Lin**? The reader can get an impression from the following result, proved in [163] and stated here without proof:

THEOREM 40. *Suppose that $L \in$* NExt **Lin** *and there is a Kripke frame of infinite length validating L. Then there exists a Kripke incomplete logic in* NExt L.

As promised at the beginning of this section, we conclude by discussing a complete solution to our original research problem relativised to the set of logics in NExt**Lin**. The proofs can be found in [162, 165].

THEOREM 41. *There are algorithms which, given a formula φ, decide whether* $\mathbf{Lin} \oplus \varphi$

- *has the finite model property;*
- *has the interpolation property;*
- *is Kripke complete;*
- *is strongly complete;*
- *is canonical.*

We shall not go into details of the proof here. But the reader can get some impression regarding the combinatorics and methods involved from the criteria which allow us to decide whether a linear tense logic is canonical and strongly complete. Denote by $\mathcal{B}_+$ the class of frames containing $\mathcal{B}$ together with frames $\mathfrak{C}(n_1, ⓚ, n_2)$ defined as follows. Suppose $k > 1$, $n_1, n_2 < \omega$ are such that $n_1 + n_2 > 0$ and $ⓚ = \{a_0, \dots, a_{k-1}\}$. Then

$$\mathfrak{C}(n_1, ⓚ, n_2) \quad = \quad \langle \omega^<(n_1) \lhd ⓚ \lhd \omega^>(n_2), P\rangle,$$

where P is the set of possible values generated by $\{X_i \mid 0 \le i \le k-1\}$, for

$$X_i \quad = \quad \{a_i\} \cup \{kj + i \mid j \in \omega\} \cup \{k^*j^* + i^* \mid j \in \omega\}$$

and $\{0^*, 1^*, \dots, n^*, \dots\}$ being the points in $\omega^>(n_2)$.

Let $\mathcal{F}$ be the class of frames of the form

$$\langle \{0, \dots, n_1\}, <, >\rangle \lhd ① \lhd \langle \{0, \dots, n_2\}, <, >\rangle \quad \text{or} \quad \langle \{0, \dots, n\}, <, >\rangle\,.$$

THEOREM 42. (i) *A logic $L \in$* NExt**Lin** *is canonical iff the underlying Kripke frame of each frame $\mathfrak{F} \in [\mathcal{B}_+^*]$ for L validates L as well.*

(ii) *A logic $L \in$* NExt**Lin** *is strongly complete iff for each frame $\mathfrak{F} \in [\mathcal{B}_+^*]$ validating L, there exists a Kripke frame $\mathfrak{G}$ for L which results from $\mathfrak{F}$ by replacing*

- *every $\mathfrak{C}(n, ⓚ)$ with $\omega^<(n)$ or $\omega^<(n) \lhd \mathfrak{H} \lhd ⓚ$, for some $\mathfrak{H} \in \mathcal{F}$, and*
- *every $\mathfrak{C}(ⓚ, n)$ with $\omega^>(n)$ or $ⓚ \lhd \mathfrak{H} \lhd \omega^>(n)$, for some $\mathfrak{H} \in \mathcal{F}$, and*
- *every $\mathfrak{C}(n_1, ⓚ, n_2)$ with $\omega^<(n_1) \lhd \mathfrak{H} \lhd \omega^>(n_2)$, for some $\mathfrak{H} \in \mathcal{F}$.*

EXAMPLE 43. The logic $\mathbf{R}_t$ of the real line is not canonical because $\mathfrak{C}(2, ②) \models \mathbf{R}_t$, but $\omega^<(2) \lhd ② \not\models \mathbf{R}_t$. However, $\mathbf{R}_t$ is strongly complete, since $\mathfrak{F} \models \mathbf{R}_t$ whenever $\mathfrak{G} \in [\mathcal{B}_+^*]$ validates $\mathbf{R}_t$ and $\mathfrak{F}$ is obtained from $\mathfrak{G}$ as in the formulation of Theorem 42 with $\mathfrak{H} = \bullet \in \mathcal{F}$.

8 SUBFRAME LOGICS

If you randomly pick a logic from the list of Modal Logic celebrities introduced in this handbook, it is very likely that its frames are closed under the *formation of subframes* (as before, a Kripke frame $\mathfrak{G} = \langle V, S_1, \dots, S_n \rangle$ is a *subframe* of a Kripke frame $\mathfrak{F} = \langle W, R_1, \dots, R_n \rangle$ if $V \subseteq W$ and $S_i = R_i \restriction V$, for $1 \leq i \leq n$). Examples are the standard unimodal logics **K** (because the class of all frames is closed under subframes), **K4**, **S5**, and **K4.3** (because transitivity, symmetry, reflexivity, and right-linearity are definable using universal first-order formulas and therefore preserved under the formation subframes), as well as **GL** and **Grz** (because the class of Noetherian frames is closed under subframes).

Moreover, many important operations on classes of frames preserve the property of being closed under forming subframes:

- the union and intersection of two classes of frames;
- the *fusion* $\mathcal{K}_1 \otimes \mathcal{K}_1$ of classes of frames $\mathcal{K}_1$ and $\mathcal{K}_2$ (see Chapter 15 which, in the unimodal case, is defined by
$$\mathcal{K}_1 \otimes \mathcal{K}_2 \quad = \quad \{\langle W, R_1, R_2 \rangle \mid \langle W, R_1 \rangle \in \mathcal{K}_1, \langle W_2, R_2 \rangle \in \mathcal{K}_2\};$$
- the *tense extension* (or addition of converse) $\mathcal{K}_t$ of a class of frames $\mathcal{K}$, where
$$\mathcal{K}_t \quad = \quad \{\langle W, R, R^{-1} \rangle \mid \langle W, R \rangle \in \mathcal{K}\};$$
- the *Boolean extension* $\mathcal{K}^{\cap,\neg}$ of a class of frames $\mathcal{K}$ which consists of all frames
$$\langle W, R_1, \dots, R_n, R_1 \cap R_2, W - R_1, \dots \rangle$$
with 2^{2^n} relations corresponding to the Boolean combinations of the R_i, where $\langle W, R_1, \dots, R_n \rangle \in \mathcal{K}$.

Using these operators we obtain numerous additional important modal 'subframe logics:' multimodal fusions like $\mathbf{S5}_n$, minimal tense extensions like $\mathbf{K4.3}_t$, and Boolean Modal Logics like $\mathbf{K}^{\cap,-}$ are all examples of modal logics determined by classes of frames closed under subframes.

In this section we explore the extent to which the restriction to 'subframe logics' leads to general 'positive' results for properties like axiomatisability, decidability, Kripke completeness, and the fmp. A systematic investigation of subframe logics in NExt **K4** was launched by Fine [45] (see also [174]). Subframe logics in NExt $\mathbf{K}_n$ as well as subframe tense and provability logics were investigated in [166, 167, 164].

We begin by observing the fact that all the 'negative' results above were obtained using logics whose frames were not closed under subframes. So one might conjecture that 'subframe logics' behave better than arbitrary modal logics. The answer to this question is 'yes' and 'no.' Yes, because indeed there are general decidability (fmp, completeness, etc.) results explaining the nice behaviour of standard 'subframe logics.' The answer is 'no,' because still one can find intuitively 'simple' subframe logics with 'bad' properties.

First, we note, however, that for subframe logics a number of otherwise separable properties of modal logics fall together; for a proof see [166] or [177]. For logics from NExt **K4** this result was first obtained by Fine [45].

THEOREM 44. *Suppose L is determined by a class of Kripke frames closed under subframes. Then the following conditions are equivalent:*

(i) $\mathsf{Fr}\, L$ *is universal,*

(ii) $\mathsf{Fr}\, L$ *is first-order definable,*

(iii) L *is $\mathcal{D}$-persistent,*

(iv) L *is strongly Kripke complete,*

(v) $\mathsf{Fr}\, L$ *has the finite embedding property—i.e., $\mathfrak{F} \in \mathsf{Fr}\, L$ iff every finite subframe of $\mathfrak{F}$ is in $\mathsf{Fr}\, L$.*

Note that for cofinal subframe logics $L \in \mathrm{NExt}\,\mathbf{K4}$ conditions (ii)–(iv) are equivalent [174].

Thomason's analysis for subframe logics

Now let us see to which extent Thomason's results (Theorem 1) hold for modal logics determined by classes of frames $\mathsf{Fr}\,\varphi$ closed under subframes.

THEOREM 45. (a) *There is an $\mathcal{ML}_6$-formula φ such that $\mathsf{Fr}\,\{\varphi\}$ is closed under subframes and the set*

$$\mathsf{Th}\,\mathsf{Fr}\,\{\varphi\} \quad = \quad \{\psi \in \mathcal{ML}_6 \mid \varphi \models\!\!\!\!\!\!\approx \psi\}$$

is Π^1_1-complete.

(b) *It is decidable whether, given $\varphi \in \mathcal{ML}_n$ with $\mathsf{Fr}\,\{\varphi\}$ closed under subframes, $\mathsf{Log}\,\mathsf{Fr}\,\{\varphi\}$ is consistent.*

(c) *There is a set Γ of $\mathcal{ML}_1$-formulas and an $\mathcal{ML}_1$-formula φ such that $\Gamma \models\!\!\!\!\!\!\approx \varphi$, but $\Delta \not\models\!\!\!\!\!\!\approx \varphi$ for any finite $\Delta \subset \Gamma$.*

(d) *Given a formula φ such that $\mathsf{Fr}\,\{\varphi\}$ is closed under subframes, every ψ with $\varphi \not\models\!\!\!\!\!\!\approx \psi$ is refutable in a countable frame validating φ.*

Claim (a) can be proved by modifying the proof of Theorem 1 (a) discussed above by introducing modalities for the converse relations of the relations R_1 and R_2 and modalities for the immediate successor relations for R_1 and R_2 and describing in $\mathcal{ML}_6$ by means of a single formula φ sufficiently many properties of the class of all subframes of the product frame $\langle \omega, <, +1\rangle \times \langle \omega, <, +1\rangle$. (Notice, however, that those subframes are not necessarily product frames).

As the reduction of multimodal logics to unimodal logics employed in the proof of Theorem 1 (a) does not preserve the property of being a subframe logic, it is an open problem whether such examples of unimodal logics exist.

Claim (c) still holds since the example provided in the proof of Theorem 1 (c) was closed under subframes. Claims (b) and (d) are now 'positive:' (b) is trivial because it is sufficient to check whether φ is valid in at least one singleton frame. (d) can be proved by constructing inductively from a frame refuting ψ a countable subframe by selecting the witnesses required to satisfy $\Diamond$-formulas; see [166] for details.

Normal subframe logics

We have not defined yet the notion of a subframe logic in a formal way because a proper definition should be slightly more general than the one suggested by considering Kripke frames only: to be able to provide a syntactic characterisation of subframe logics, to ensure that $\mathbf{K}_n \oplus \Sigma_1 \oplus \Sigma_2$ is a subframe logic whenever both $\mathbf{K}_n \oplus \Sigma_1$ and $\mathbf{K}_n \oplus \Sigma_2$ are

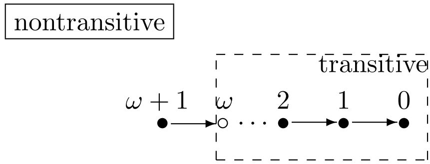


Figure 4.

subframe logics, and finally, to cover a number of interesting Kripke incomplete logics (like bimodal provability logics), we extend the definition of a subframe to general frames. Recall that a frame $\mathfrak{G} = \langle V, S_1, \dots, S_n, P \rangle$ is a *subframe* of a frame $\mathfrak{F} = \langle W, R_1, \dots, R_n, Q \rangle$ if $V \subseteq W$, $P = \{A \cap V \mid A \in Q\}$, and $S_i = R_i \restriction V$, for $1 \le i \le n$. A logic L is a *subframe logic* if its class of general frames is closed under subframes.

To describe subframe logics syntactically, define inductively the *relativisation* φ^p of a formula φ to a propositional variable p (which does not occur in φ) by taking

$$\begin{array}{lcl} q^p & = & q \wedge p, \quad q \text{ an atom}, \\ (\psi \odot \chi)^p & = & \psi^p \odot \chi^p, \quad \text{for } \odot \in \{\wedge, \vee, \rightarrow\}, \\ (\Box_i \psi)^p & = & \Box_i (p \rightarrow \psi^p) \wedge p \end{array}$$

and put $\varphi^{sf} = p \rightarrow \varphi^p$. It is not difficult to show that φ^{sf} is valid in a frame $\mathfrak{F}$ iff φ is valid in all subframes of $\mathfrak{F}$. (Notice that if $\mathfrak{M}$ is a model based on $\mathfrak{F}$, $\mathfrak{M}'$ a model based on the subframe of $\mathfrak{F}$ induced by $\{y \mid (\mathfrak{M}, y) \models p\}$ and $(\mathfrak{M}, x) \models q$ iff $(\mathfrak{M}', x) \models q$, for all variables q, then $(\mathfrak{M}, x) \models \varphi^p$ iff $(\mathfrak{M}', x) \models \varphi$.) Therefore, we obtain

PROPOSITION 46. *The following conditions are equivalent for any $L \in \mathrm{NExt}\,\mathbf{K}_n$:*

(i) *L is a subframe logic,*

(ii) *$L = \mathbf{K}_n \oplus \{\varphi^{sf} \mid \varphi \in \Gamma\}$, for some set of formulas Γ,*

(iii) *L is characterised by a class of frames closed under subframes.*

Based on this proposition, it is not difficult to see that the class of subframe logics forms a complete sublattice of $\mathrm{NExt}\,\mathbf{K}_n$. Now, the first question to address is whether there are Kripke incomplete subframe logics at all:

EXAMPLE 47. (Van Benthem 1979) Let $\mathfrak{F} = \langle W, R, P \rangle$ be the frame whose underlying Kripke frame is shown in Fig. 4 ($\omega + 1$ sees only ω and the subframe generated by ω is transitive) and $X \subseteq W$ is in P iff either X is finite and $\omega \notin X$ or X is cofinite in W and $\omega \in X$. It is easy to see that P is closed under $\cap$, $-$ and $\Diamond$.

Let L be the logic of the frame $\mathfrak{F}$ constructed in Example 47. Since every rooted subframe $\mathfrak{G}$ of $\mathfrak{F}$ is isomorphic to a generated subframe of $\mathfrak{F}$, L is a subframe logic. We show that L has the same Kripke frames as $\mathbf{GL.3}$. Suppose $\mathfrak{G}$ is a rooted Kripke frame for $\mathbf{GL.3}$ refuting $\varphi \in L$. Then clearly $\mathfrak{G}$ contains a finite subframe $\mathfrak{H}$ refuting φ. Since $\mathfrak{H}$ is a finite chain of irreflexive points, it is isomorphic to a generated subframe of $\mathfrak{F}$, contrary to $\mathfrak{F} \not\models \varphi$. Thus $\mathfrak{G} \models L$. Conversely, suppose $\mathfrak{G}$ is a Kripke frame for L. Then $\mathfrak{G}$ is irreflexive. For otherwise $\mathfrak{G}$ refutes the formula $\varphi = \Box^2(\Box p \rightarrow p) \wedge \Box(\Box p \rightarrow p) \rightarrow \Box p$, which is valid in $\mathfrak{F}$. Let us show now that $\mathfrak{G}$ is transitive. Suppose otherwise. Then $\mathfrak{G}$ refutes the formula $\Box p \rightarrow \Box(\Box p \vee (\Box q \rightarrow q))$, which is valid in $\mathfrak{F}$ because ω is a reflexive point. Finally, since $\mathfrak{G} \models \varphi$, $\mathfrak{G}$ is Noetherian and since $\mathfrak{F}$ is of width 1, we may conclude that $\mathfrak{G} \models \mathbf{GL.3}$. It follows that the subframe logic L is Kripke incomplete. Indeed, it shares the same class of Kripke frames with $\mathbf{GL.3}$ but $\Box p \rightarrow \Box\Box p \in \mathbf{GL.3} - L$.

We now follow the path described above for NExt $\mathbf{K}_n$ and analyse the location of Kripke complete and decidable subframe logics within the lattice of subframe logics.

Subframe logics of frames with transitive relations

A number of surprisingly general decidability and completeness results have been obtained for subframe logics whose frames are based on transitive relations. The first fundamental result on the finite model property (and therefore decidability and Kripke completeness) of subframe logics is due to Fine [45]. Although the following theorem can be obtained for free using the machinery of canonical formulas, here we show its proof using the *maximal points technique* of [43].

THEOREM 48 (Fine). *All unimodal subframe logics of transitive general frames (i.e., extensions of* **K4**) *have the finite model property. All finitely axiomatisable ones are decidable.*

Proof. The proof is based on the following fundamental observation: call a point x in a transitive frame $\mathfrak{F} = \langle W, R, P\rangle$ *non-eliminable* if there is $X \in P$ such that $x \in X$ but no proper successor of x is in X (in other words, x is *maximal* in X); in this case we write $x \in \max_R X$. Then one can show the following: if $\mathfrak{F}$ is descriptive and $x \in X \in P$, then there exists a point $y \in \max_R X$ such that $x = y$ or xRy.

Now suppose that $L \supseteq \mathbf{K4}$ is a subframe logic. To prove that L has the finite model property, suppose that $\varphi \notin L$. Take a model $\mathfrak{M} = \langle \mathfrak{F}, \mathfrak{V}\rangle$ refuting φ at a point x and based on a descriptive frame $\mathfrak{F} = \langle W, R, P\rangle$ for L. Say that a point $x \in W$ is *non-eliminable* relative to φ if there is a subformula ψ of φ such that $x \in \max_R\{y \in W \mid y \models \psi\}$ or $x \in \max_R\{y \in W \mid y \models \neg\psi\}$. Now select recursively a set of points W^φ as follows:

- Set $V = \{x\}$.
- If there exist $y \in V$ and $\Box\psi \in \mathsf{sub}\,\varphi$ such that $y \models \neg\Box\psi$, and there does not exist $z \in V$ with $z \models \neg\psi$ and yRz, then set $V = V \cup \{z\}$ for some non-eliminable z (relative to φ) with yRz and $z \models \neg\psi$. Otherwise, set $W^\varphi = V$ and stop.

Construct a new model $\mathfrak{M}^\varphi$ based on the frame $\mathfrak{F}^\varphi = \langle W^\varphi, R{\restriction}W^\varphi\rangle$ by taking $\mathfrak{V}^\varphi(p) = \mathfrak{V}(p) \cap W^\varphi$ for all variables p in φ. Clearly, the Kripke frame $\mathfrak{F}^\varphi$ is rooted, of depth $\leq l(\varphi)$, and no point has more than $\ell(\varphi)$ successors. Besides, one can easily show that $(\mathfrak{M}^\varphi, y) \models \psi$ iff $(\mathfrak{M}, y) \models \psi$, for all $\psi \in \mathsf{sub}\,\varphi$ and $y \in W^\varphi$. Finally, one can show that $\mathfrak{F}^\varphi$ is a frame for L (this is trivial if L is $\mathcal{D}$-persistent; considerably more work is required if L is not $\mathcal{D}$-persistent.) ❑

Let us consider now what happens if we move from a unimodal subframe logic $L \in$ NExt **K4** to the tense logic determined by $(\mathsf{Fr}\,L)_t$. In other words, we move from unimodal logics to bimodal logics, where the second operator is interpreted by the inverse of the accessibility relation of the first operator. From a syntactic viewpoint, this semantic condition is captured by the axioms

$$p \to \Box_1\Diamond_2 p \qquad \text{and} \qquad p \to \Box_2\Diamond_1 p.$$

So, we set, for any normal unimodal logic $L = \mathbf{K} \oplus \Gamma$,

$$L_t \quad = \quad \mathbf{K}_2 \;\oplus\; \Gamma \;\oplus\; \{p \to \Box_1\Diamond_2 p, p \to \Box_2\Diamond_1 p\}$$

and call L the *minimal tense extension* of L. General questions regarding minimal tense extensions are: What is the relation between L_t and the logic determined by the class of frames $(\mathsf{Fr}\,L)_t$? Are Kripke completeness and decidability inherited from unimodal logics to minimal tense extensions? As far as Kripke completeness is concerned, the answer is negative: [161] gives an example of a logic $L \in \mathrm{NExt}\,\mathbf{K4}$ with the finite model property such that L_t is not Kripke complete. Transfer of decidability is an open problem. However, for minimal tense extensions of subframe logics in $\mathrm{NExt}\,\mathbf{K4}$ a complete answer has been given in [160, 164]:

THEOREM 49. *If $L \in \mathrm{NExt}\mathbf{K4}$ is a subframe logic then*
(i) *L_t is Kripke complete and determined by $(\mathsf{Fr}\,L)_t$;*
(ii) *L_t has the finite model property iff L is canonical iff $\mathsf{Fr}\,L$ is first-order definable;*
(iii) *L_t is decidable whenever L is finitely axiomatisable.*

So, by (i), L_t is indeed Kripke complete whenever L is a subframe logic containing **K4**. The first exciting bit of this theorem is (ii). It connects the finite model property of minimal tense extensions with first-order definability: for first-order definable subframe logics like **K4**, **S4**, **K4.3**, and **S4.3** we obtain that their minimal tense extensions still have the finite model property, while minimal tense extensions of subframe logics that are not first-order definable, say, $\mathbf{GL}_t$ and $\mathbf{Grz}_t$, do not enjoy this property. For some examples, such as $\mathbf{Grz}_t$, this can be proved easily: just observe that the Grzegorczyk axiom

$$\Box_2(\Box_2(p \to \Box_2 p) \to p) \to p$$

is refuted in $\langle \omega, \geq, \leq \rangle$ and so does not belong to $\mathbf{Grz}_t$; however, it is clearly valid in all finite partial orders (which coincide with the finite frames in $(\mathsf{Fr}\,\mathbf{Grz})_t$).

The second interesting news here is that (iii) provides us with a decidability result for a class of logics which (in general) neither have the finite model property nor the tree model property. In fact, to prove (iii) [164] introduces so-called quasi-frames, that is, frames which come together with type assignments (similar to those in Section 6.2) and with respect to which minimal tense extensions of subframe logics have the bounded finite model property.

The decidability result in Theorem 49 does not cover Kripke incomplete logics, and one may wonder whether there exist at all natural and interesting classes of *decidable* modal logics containing Kripke incomplete ones. The answer is 'yes:' one such class is the set of all finitely axiomatisable (possibly non-normal) subframe logics containing the bimodal provability logic

$$\mathbf{CSM}_0 \quad = \quad \mathbf{GL} \otimes \mathbf{GL} \oplus \{\Box_1 \to \Box_2 p, \Box_2 p \to \Box_1 \Box_2 p\}$$

(named so in [159] after Carlson, Smorynski and Montagna). Almost all bimodal provability logics discussed and investigated in the literature [112, 137] are indeed (sometimes non-normal) subframe logics. For the proof of this result see [167].

The lattice of all subframe logics

We have seen above a lot of beautiful and useful decidability results for subframe logics over transitive relations. Unfortunately, the situation changes drastically as soon as not all accessibility relations are transitive. In this part we summarise what is known in this case. First we shall see that at least the 'upper part' of the lattice of subframe logics is

quite well-behaved. Recall that $\mathbf{K} \oplus \boldsymbol{alt}_n$ is the logic determined by the frames in which no point has more than n successors. The following result is proved by first showing that all subframe logics containing $\mathbf{K} \oplus \boldsymbol{alt}_n$ are Kripke complete [1] and then proving that if a formula φ is refuted in a frame validating $\boldsymbol{alt}_n$ then one can always select finitely many points from that frame in which φ is refuted. (It should be clear from the proof that this result still holds for multimodal subframe logics in which for each operator an axiom of the form $\boldsymbol{alt}_n$ holds.)

THEOREM 50. *All subframe logics containing some* $\mathbf{K} \oplus \boldsymbol{alt}_n$ *have the finite model property.*

Now we say that a subframe logic L is *strictly sf-complete* if there does not exist another subframe logic L' with the same Kripke frames as L. In comparison with NExt $\mathbf{K}$, where according to Blok's dichotomy for almost every interesting L there exist uncountably many modal logic L' with the same Kripke frames as L, the situation is much more diverse in the lattice of subframe logics. Example 47 shows that **GL.3** is not strictly sf-complete. However, the logics **T**, **S4** and **Grz** turn out to be strictly sf-complete. As in the full lattice of normal modal logics, the notion of strict sf-completeness is closely related to the notion of splittings (now in the lattice of subframe logics) and the decidability of the axiomatisation problem for formulas axiomatising subframe logics. Say that a subframe logic $L \in$ NExt $\mathbf{K}$ is a *subframe union-splitting* by a set $\mathcal{F}$ of finite rooted frames, in symbols

$$L \quad = \quad \mathbf{K}/^{sf}\mathcal{F},$$

if L is the smallest subframe logic such that $\mathfrak{F} \not\models L$, for at least one $\mathfrak{F} \in \mathcal{F}$. For example,

$$\mathbf{T} \quad = \quad \mathbf{K}/^{sf} \circ .$$

It is now readily checked that any Kripke complete subframe logic L which is a subframe union-splitting by finitely many rooted frames is strictly sf-complete and that the set $\{\varphi \mid \mathbf{K} \oplus \varphi^{sf} = L\}$ is recursive whenever L is decidable. Based on these observations, the following partial results were obtained in [166]:

THEOREM 51. (a) *A subframe logic L containing* **K4** *is strictly sf-complete iff $L \not\subseteq$* **GL.3** *iff L is a subframe union-splitting. Moreover, if $L \not\subseteq$* **GL.3** *is finitely axiomatisable, then $\{\varphi \mid \mathbf{K} \oplus \varphi^{sf} = L\}$ is recursive.*

(b) *All subframe logics $L \in$ NExt$\mathbf{K} \oplus \boldsymbol{alt}_n$ are strictly sf-complete and subframe union-splittings. Moreover, if L is finitely axiomatisable, then $\{\varphi \mid \mathbf{K} \oplus \varphi^{sf} = L\}$ is recursive.*

(c) *It is decidable whether a finitely axiomatised subframe logic is determined by a finite number of finite frames* (*is tabular*).

No general *undecidability results* have been proved so far for subframe logics.

PROBLEM 15. Characterise the class of subframe logics L for which $\{\varphi \mid \mathbf{K} \oplus \varphi^{sf} = L\}$ is recursive.

For example, although **K4** is not a subframe union-splitting it is an open question whether $\{\varphi \mid \mathbf{K} \oplus \varphi^{sf} = \mathbf{K4}\}$ is recursive.

Cl	$=$	$\mathbf{Int} + p \vee \neg p$
SmL	$=$	$\mathbf{Int} + (\neg q \to p) \to (((p \to q) \to p) \to p)$
KC	$=$	$\mathbf{Int} + \neg p \vee \neg\neg p$
LC	$=$	$\mathbf{Int} + (p \to q) \vee (q \to p)$
SL	$=$	$\mathbf{Int} + ((\neg\neg p \to p) \to \neg p \vee p) \to \neg p \vee \neg\neg p$

Table 3. A list of standard superintuitionistic logics

9 SUPERINTUITIONISTIC LOGICS

Although C.I. Lewis constructed his first modal calculus **S3** in 1918, it was Gödel's [52] two page note that attracted serious attention of mathematical logicians to modal systems. While Lewis [90] used an abstract necessity operator to avoid paradoxes of the material implication, Gödel and earlier Orlov [117][7] treated $\Box$ as 'it is provable' to give a classical interpretation of intuitionistic propositional logic **Int** of Brouwer [12, 13] and Heyting [63] by means of embedding it into a modal 'provability' system which turned out to be equivalent to Lewis' **S4**.

Approximately at the same time Gödel [51] observed that there are infinitely many logics located between **Int** and classical logic **Cl**, which—together with the creation of constructive (proper) extensions of **Int** by Kleene [74] and Rose [122] (realisability logic), Medvedev [109] (logic of finite problems), Kreisel and Putnam [78]—gave an impetus to studying the class of logics intermediate between **Int** and **Cl**, started by Umezawa [149, 150]. Gödel's embedding of **Int** into **S4**, presented in an algebraic form by McKinsey and Tarski [108] and extended to all intermediate logics by Dummett and Lemmon [33], made it possible to develop theories of modal and intermediate logics in parallel ways. And the structural results of Blok [9] and Esakia [36, 36] establishing an isomorphism between the lattices Ext **Int** and NExt **Grz**, along with preservation results of Maksimova and Rybakov [100] and Zakharyaschev [171], transferring various properties from modal to intermediate logics and back, showed that in many respects the theory of intermediate logics is reducible to the theory of logics in NExt **S4**.

To demonstrate this as well as some features of superintuitionistic logics is the main aim of this section. We will use the same system of notation as in the modal case. In particular, Ext **Int** is the lattice of all logics of the form $\mathbf{Int} + \Gamma$ (where Γ is an arbitrary set of formulas in the language of **Int** and $+$ means taking the closure under *modus ponens* and substitution); we call them *superintuitionistic logics* or *si-logics* for short. Basic facts about the syntax and semantics of **Int** and relevant references can be found in [157, 24]. A list of some 'standard' si-logics is given in Table 3.

9.1 Intuitionistic frames

As in the case of modal logics, the adequate relational semantics for si-logics can be constructed on the base of the Stone representation of the algebraic 'models' for **Int**, known as *Heyting* (or *pseudo-Boolean*) *algebras*. It is hard to trace now who was the first to introduce intuitionistic general frames—the earliest references we know are [35] and

[7] Orlov's paper remained unnoticed till the end of the 1980s. It is remarkable also for constructing the first system of relevant logic.

[119]—but in any case, having at hand [71] and [54], the construction must have been clear.

An *intuitionistic* (*general*) *frame* is a triple $\mathfrak{F} = \langle W, R, P\rangle$ in which R is a partial order on $W \neq \emptyset$ and P is a collection of upward closed subsets (cones) in W containing $\emptyset$ and closed under $\cap$, $\cup$, and the operation $\supset$ (for $\to$) defined by

$$X \supset Y \quad = \quad \{x \in W \mid \forall y \in x\!\uparrow (y \in X \to y \in Y)\}.$$

If P contains all upward closed subsets of W then $\mathfrak{F}$ is called an *intuitionistic Kripke frame* and denoted by $\mathfrak{F} = \langle W, R\rangle$. An important feature of intuitionistic models $\mathfrak{M} = \langle \mathfrak{F}, \mathfrak{V}\rangle$ (where $\mathfrak{V}$ maps propositional variables to sets in P) is that $\mathfrak{V}(\varphi)$, the value of a formula φ in $\mathfrak{M}$, is always upward closed. Every intuitionistic frame $\mathfrak{F} = \langle W, R, P\rangle$ gives rise to the Heyting algebra $\mathfrak{F}^+ = \langle P, \cap, \cup, \supset, \emptyset\rangle$ called the *dual* of $\mathfrak{F}$. Conversely, given a Heyting algebra $\mathfrak{A} = \langle A, \wedge, \vee, \to, \bot\rangle$, we construct its relational representation $\mathfrak{A}_+ = \langle W, R\rangle$ by taking W to be the set of all prime filters in $\mathfrak{A}$ (a filter ∇ is *prime* if it is proper and $a \vee b \in \nabla$ implies $a \in \nabla$ or $b \in \nabla$), R to be the set-theoretic inclusion $\subseteq$ and

$$P \quad = \quad \{\{\nabla \in W \mid a \in \nabla\} \mid a \in A\}.$$

It is readily checked that $\mathfrak{A}_+$, the *dual* of $\mathfrak{A}$, is an intuitionistic frame and $\mathfrak{A} \cong (\mathfrak{A}_+)^+$. A frame $\mathfrak{F}$ is called *descriptive* if $\mathfrak{F} \cong (\mathfrak{F}^+)_+$. Duality between the basic truth-preserving operations on algebras and descriptive frames (taking p-morphic images, generated subframes and disjoint unions) is established by the same technique as in the modal case.

Since every consistent si-logic L is characterised by its Tarski–Lindenbaum algebra $\mathfrak{A}_L$, we conclude that L is also characterised by a class of intuitionistic frames, say, by the dual of $\mathfrak{A}_L$.

At the algebraic level, the connection between **Int** and **S4** discovered by Gödel is reflected by the fact, established in [107], that the algebra of open elements (i.e., elements a such that $\Box a = a$) of every modal algebra for **S4** (known as a *topological Boolean algebra*; see [118]) is a Heyting algebra and, conversely, every Heyting algebra is isomorphic to the algebra of open elements of a suitable algebra for **S4**. We explain this result in the frame-theoretic language.

Given a frame $\mathfrak{F} = \langle W, R, P\rangle$ for **S4** (which means that R is a quasi-order on W), we denote by $\boldsymbol{\rho} W$ the set of clusters in $\mathfrak{F}$—more generally, $\boldsymbol{\rho} X = \{C(x) \mid x \in X\}$—and put $C(x)\boldsymbol{\rho} C(y)$ iff xRy,

$$\boldsymbol{\rho} P \quad = \quad \{\boldsymbol{\rho} X \mid\ X \in P \,\&\, X = \Box X\} \quad = \quad \{\boldsymbol{\rho} X \mid X \in P \wedge X \,\&\, X\!\uparrow\}.$$

It is readily checked that $\boldsymbol{\rho}\mathfrak{F} = \langle \boldsymbol{\rho} W, \boldsymbol{\rho} R, \boldsymbol{\rho} P\rangle$ is an intuitionistic frame; we call it the *skeleton* of $\mathfrak{F}$. The *skeleton* of a model $\mathfrak{M} = \langle \mathfrak{F}, \mathfrak{V}\rangle$ for **S4** is the model $\boldsymbol{\rho}\mathfrak{M} = \langle \boldsymbol{\rho}\mathfrak{F}, \boldsymbol{\rho}\mathfrak{V}\rangle$, where $\boldsymbol{\rho}\mathfrak{V}(p) = \mathfrak{V}(\Box p)$.

Denote by T the *Gödel translation* prefixing $\Box$ to all subformulas of a given intuitionistic formula.[8] Then for every model $\mathfrak{M}$ for **S4**, every intuitionistic formula φ and every point x in $\mathfrak{M}$,

$$(\boldsymbol{\rho}\mathfrak{M}, C(x)) \models \varphi \qquad \text{iff} \qquad (\mathfrak{M}, x) \models T(\varphi).$$

It follows that $\varphi \in$ **Int** implies $T(\varphi) \in$ **S4**. To prove the converse, we should be able to convert intuitionistic frames $\mathfrak{F}$ into modal ones with the skeleton (isomorphic to) $\mathfrak{F}$.

[8]The translation defined in [52] does not prefix $\Box$ to conjunctions and disjunctions. However, this difference is of no importance as far as embeddings into logics in NExt**S4** are concerned.

This is trivial if $\mathfrak{F}$ is a Kripke frame—we can just regard it to be a frame for **S4**, which in view of the Kripke completeness of both **Int** and **S4**, shows that T really embeds the former into the latter, i.e.,

$$\varphi \in \mathbf{Int} \qquad \text{iff} \qquad T(\varphi) \in \mathbf{S4}.$$

In general, one can construct a modal frame from an intuitionistic frame $\mathfrak{F} = \langle W, R, P\rangle$ by taking the closure $\boldsymbol{\sigma} P$ of P under the Boolean operations. Then $\langle W, R, \boldsymbol{\sigma} P\rangle$ is a partially ordered modal frame; we denote it by $\boldsymbol{\sigma}\mathfrak{F}$. Moreover, we clearly have $\mathfrak{F} \cong \boldsymbol{\rho\sigma}\mathfrak{F}$. It is worth noting that if $\mathfrak{F} = \langle W, R\rangle$ is a finite intuitionistic Kripke frame then $\boldsymbol{\sigma}\mathfrak{F}$ is also a Kripke frame. However, for an infinite $\mathfrak{F}$, $\boldsymbol{\sigma}\mathfrak{F}$ is not in general a Kripke frame—witness $\langle \omega, \leq\rangle$.

9.2 Canonical formulas

The language of canonical formulas, axiomatising all si-logics and characterising the structure of their frames, can be easily developed following the scheme of constructing the canonical formulas for **K4** and using the connection between modal and intuitionistic frames established above. We confine ourselves here only to pointing out the differences from the modal case and some interesting peculiarities; details can be found in [169, 170, 24]. Actually, there are two important differences. First, in the definition of subreduction of $\mathfrak{F} = \langle W, R, P\rangle$ to $\mathfrak{G}$ the third condition does not correspond to the fact that all sets in P are upward closed. We replace it by the following condition

- $\forall X \in \overline{Q}\ f^{-1}(X){\downarrow} \in \overline{P}$, where $\overline{Q} = \{V - X \mid X \in Q\}$ and $\overline{P} = \{W - X \mid X \in P\}$.

$\mathfrak{G}$ is a *subframe* of $\mathfrak{F}$ if there is an injective subreduction of $\mathfrak{F}$ to $\mathfrak{G}$. It is of interest to note that in the intuitionistic case (cofinal) subreductions are dual to IC(N)-subalgebras of Heyting algebras which only preserve implication, conjunction (and negation) but do not necessarily preserve disjunction.

Second, we have to change the definition of open domains. Now we say an antichain $\mathfrak{a}$ (of at least two points) is an *open domain* in an intuitionistic model $\mathfrak{N}$ relative to a formula φ if there ia a pair $t_{\mathfrak{a}} = (\Gamma_{\mathfrak{a}}, \Delta_{\mathfrak{a}})$ such that $\Gamma_{\mathfrak{a}} \cup \Delta_{\mathfrak{a}} = \mathsf{sub}\,\varphi$, $\bigwedge \Gamma_{\mathfrak{a}} \to \bigvee \Delta_{\mathfrak{a}} \notin \mathbf{Int}$ and

- $\psi \in \Gamma_{\mathfrak{a}}$ iff $a \models \psi$ for all $a \in \mathfrak{a}$.

It is worth noting that in any intuitionistic model every antichain $\mathfrak{a}$ is open relative to every disjunction free formula φ. Indeed, let $\Gamma_{\mathfrak{a}}$ be defined by condition above and $\Delta_{\mathfrak{a}} = \mathsf{sub}\,\varphi - \Gamma_{\mathfrak{a}}$. It should be clear that $\psi \wedge \chi \in \Gamma_{\mathfrak{a}}$ iff $\psi \in \Gamma_{\mathfrak{a}}$ and $\chi \in \Gamma_{\mathfrak{a}}$. And if $\psi \to \chi \in \Gamma_{\mathfrak{a}}$, $\psi \in \Gamma_{\mathfrak{a}}$ but $\chi \in \Delta_{\mathfrak{a}}$ then $a \models \psi$ for every $a \in \mathfrak{a}$ and $b \not\models \chi$ for some $b \in \mathfrak{a}$, whence $b \not\models \psi \to \chi$, which is a contradiction. It follows that $\bigwedge \Gamma_{\mathfrak{a}} \to \bigvee \Delta_{\mathfrak{a}} \notin \mathbf{Int}$.

Now, as in the modal case, with every finite rooted intuitionistic frame $\mathfrak{F} = \langle W, R\rangle$ and a set $\mathfrak{D}$ of antichains in it we can associate two formulas $\beta(\mathfrak{F}, \mathfrak{D}, \bot)$ and $\beta(\mathfrak{F}, \mathfrak{D})$, called the *canonical* and *negation free canonical formulas*, respectively, so that $\mathfrak{G} \not\models \beta(\mathfrak{F}, \mathfrak{D}, \bot)$ ($\mathfrak{G} \not\models \beta(\mathfrak{F}, \mathfrak{D})$) iff there is a (cofinal) subreduction of $\mathfrak{G}$ to $\mathfrak{F}$ satisfying (CDC) for $\mathfrak{D}$.

THEOREM 52. *There is an algorithm which, given an intuitionistic φ, returns canonical formulas $\beta(\mathfrak{F}_1, \mathfrak{D}_1, \bot), \dots, \beta(\mathfrak{F}_n, \mathfrak{D}_n, \bot)$ such that*

$$\mathbf{Int} + \varphi \quad = \quad \mathbf{Int} + \beta(\mathfrak{F}_1, \mathfrak{D}_1, \bot) + \cdots + \beta(\mathfrak{F}_n, \mathfrak{D}_n, \bot).$$

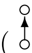
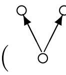
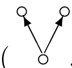
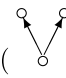
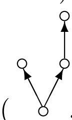
Cl	=	**Int** $+ \beta($  $)$
SmL	=	**Int** $+ \beta($  $) + \beta($  $)$
KC	=	**Int** $+ \beta($  $, \bot)$
LC	=	**Int** $+ \beta($  $)$
SL	=	**Int** $+ \beta^\sharp($  $, \bot)$

Table 4. Canonical axioms of standard superintuitionistic logics

If φ is negation free then one can use only negation free canonical formulas. And if φ is disjunction free then all the $\mathfrak{D}_i$ are empty.

In the intuitionistic case we have the following syntactical characterisation of subframe and cofinal subframe si-logics:

THEOREM 53. (1) *A si-logic is axiomatisable by implicative formulas iff it is determined by a class of* (*finite*) *frames closed under subframes.*

(2) *A si-logic is axiomatisable by disjunction-free formulas iff it is determined by a class of* (*finite*) *frames closed under cofinal subframes.*

It follows from these two theorems (and the refutability criterion for the canonical formulas) that all cofinal subframe si-logics have the finite model property and are decidable if finitely axiomatisable. In fact, there are a continuum of subframe logics [173]. That all si-logics with disjunction-free axioms have the fmp was first proved by McKay [104] with the help of Diego's [31] theorem according to which there are only finitely many pairwise nonequivalent in **Int** disjunction free formulas in variables $p_1, \ldots, p_n$ (see also [151]). An algebraic approach to superintuitionistic (cofinal) subframe logics has been recently developed in [2].

PROBLEM 16. Characterise the computational complexity of cofinal subframe si-logics.

Table 4 shows canonical axiomatisations of the si-logics in Table 3. Using this 'geometrical' representation it is not hard to see, for instance, that **SmL**, known as the *Smetanich logic*, is the greatest consistent extension of **Int** different from **Cl**; it is the logic of the two-point rooted frame. **KC**, the logic of the *Weak Law of the Excluded Middle*, is characterised by the class of directed frames. It is the greatest si-logic containing the same negation free formulas as **Int** [68]. **LC**, the *Dummett* or *chain logic*, is characterised by the class of linear frames [32].

Jankov [69] proved that logics of the form **Int**$+\beta(\mathfrak{F}, \mathfrak{D}^\sharp, \bot)$ and only them are splittings of Ext **Int**. However, not every si-logic is a union-splitting of Ext **Int**.

PROBLEM 17. Characterise the degree of Kripke incompleteness of si-logics.

9.3 Modal companions and preservation theorems

The fact that the Gödel translation T embeds **Int** into **S4** and the relationship between intuitionistic and modal frames shown above can be used to reduce various problems concerning **Int** (e.g., proving decidability, Kripke completeness or the fmp) to those for **S4** and vice versa. Moreover, it turns out that each logic in Ext **Int** is embedded by T into some logics in NExt **S4**, and for each logic in NExt **S4** there is a logic in Ext **Int** embeddable in it.

We say a modal logic $M \in$ NExt **S4** is a *modal companion* of a si-logic L if L is embedded in M by T, i.e., if for every intuitionistic formula φ, we have $\varphi \in L$ iff $T(\varphi) \in M$. If M is a modal companion of L then L is called the *si-fragment* of M and denoted by $\boldsymbol{\rho} M$. The reason for denoting the operator 'modal logic $\mapsto$ its si-fragment' by the same symbol we used for the skeleton operator is explained by the simple fact that $\boldsymbol{\rho} M = \{\varphi \mid T(\varphi) \in M\}$ and, moreover, if M is characterised by a class $\mathcal{C}$ of modal frames then $\boldsymbol{\rho} M$ is characterised by the class $\boldsymbol{\rho}\mathcal{C} = \{\boldsymbol{\rho}\mathfrak{F} \mid \mathfrak{F} \in \mathcal{C}\}$ of intuitionistic frames.

Thus, $\boldsymbol{\rho}$ maps NExt**S4** into Ext**Int**. The following observation of [33] shows that actually $\boldsymbol{\rho}$ is a surjection. Given a logic $L \in$ Ext **Int**, let

$$\boldsymbol{\tau} L \quad = \quad \mathbf{S4} \oplus \{T(\varphi) \mid \varphi \in L\}.$$

Then, for every si-logic L, $\boldsymbol{\tau} L$ is a modal companion of L.

Now we use the language of canonical formulas to show a general characterisation of all modal companions of a given si-logic L obtained in [170, 171]. Notice first that for every modal frame $\mathfrak{G}$ and every intuitionistic canonical formula $\beta(\mathfrak{F}, \mathfrak{D}, \bot)$, $\mathfrak{G} \models \alpha(\mathfrak{F}, \mathfrak{D}, \bot)$ iff $\boldsymbol{\rho}\mathfrak{G} \models \beta(\mathfrak{F}, \mathfrak{D}, \bot)$, and so $\mathbf{S4} \oplus T(\beta(\mathfrak{F}, \mathfrak{D}, \bot)) = \mathbf{S4} \oplus \alpha(\mathfrak{F}, \mathfrak{D}, \bot)$. The same holds for the negation free canonical formulas.

THEOREM 54. *$M \in$ NExt **S4** is a modal companion of $L = \mathbf{Int} + \{\beta(\mathfrak{F}_i, \mathfrak{D}_i, \bot) \mid i \in I\}$ iff M can be represented in the form*

$$M \quad = \quad \mathbf{S4} \quad \oplus \quad \{\alpha(\mathfrak{F}_i, \mathfrak{D}_i, \bot) \mid i \in I\} \quad \oplus \quad \{\alpha(\mathfrak{F}_j, \mathfrak{D}_j, \bot) \mid j \in J\},$$

where every frame $\mathfrak{F}_j$, for $j \in J$, contains a proper cluster.

Thus, we have:

$$\boldsymbol{\rho}\mathbf{S4} \;=\; \boldsymbol{\rho}\mathbf{S4.1} \;=\; \boldsymbol{\rho}\mathbf{Dum} \;=\; \boldsymbol{\rho}\mathbf{Grz} \;=\; \mathbf{Int},$$

$$\boldsymbol{\rho}\mathbf{S4.2} \;=\; \boldsymbol{\rho}(\mathbf{S4.2} \oplus \mathbf{Grz}) \;=\; \mathbf{KC},$$

$$\boldsymbol{\rho}\mathbf{S4.3} \;=\; \boldsymbol{\rho}(\mathbf{S4.3} \oplus \mathbf{Grz}) \;=\; \mathbf{LC},$$

$$\boldsymbol{\rho}\mathbf{S5} \;=\; \boldsymbol{\rho}(\mathbf{S5} \oplus \mathbf{Grz}) \;=\; \mathbf{Cl}.$$

COROLLARY 55. *The set of modal companions of every consistent si-logic L forms the interval*

$$\boldsymbol{\rho}^{-1}(L) \quad = \quad [\boldsymbol{\tau} L, \boldsymbol{\tau} L \oplus \alpha(②)] \quad = \quad \{M \in \text{NExt}\,\mathbf{S4} \mid \boldsymbol{\tau} L \subseteq M \subseteq \boldsymbol{\tau} L \oplus \mathbf{Grz}\}$$

and contains an infinite descending chain of logics.

Proof. Notice first that $\alpha(\mathfrak{F}, \mathfrak{D}, \bot)$ and $\alpha(\mathfrak{F}, \mathfrak{D})$ are in **Grz** iff $\mathfrak{F}$ contains a proper cluster. So $\boldsymbol{\rho}^{-1}(L) \subseteq [\boldsymbol{\tau} L, \, \boldsymbol{\tau} L \oplus \alpha(②)]$. On the other hand, the si-fragments of all logics

Property of logics	Preserved under		
	$\boldsymbol{\rho}$	$\boldsymbol{\tau}$	$\boldsymbol{\sigma}$
Decidability	Yes	Yes	Yes
Kripke completeness	Yes	Yes	No
Finite model property	Yes	Yes	Yes
Tabularity	Yes	No	Yes
Interpolation property	Yes	No	No
First-order definability	Yes	Yes	No

Table 5. Preservation theorems

in the interval are the same, namely L. Therefore, $\boldsymbol{\rho}^{-1}(L) = [\boldsymbol{\tau}L, \boldsymbol{\tau}L \oplus \alpha(②)]$. Now, if L is consistent then $\beta(\circ) \notin L$ and so

$$\boldsymbol{\tau}L \subset \ldots \subset \boldsymbol{\tau}L \oplus \alpha(ⓚ) \subset \cdots \subset \boldsymbol{\tau}L \oplus \alpha(②) \subset \boldsymbol{\tau}L \oplus \alpha(①) = \mathbf{For},$$

where **For** is the set of all intuitionistic formulas. ❑

This result is due to Maksimova and Rybakov [100], Blok [9] and Esakia [37]. Thus, all modal companions of every si-logic L are contained between the least companion $\boldsymbol{\tau}L$ and the greatest one, viz., $\boldsymbol{\tau}L \oplus \alpha(②)$, which will be denoted by $\boldsymbol{\sigma}L$.

The following theorem, which is also a consequence of Theorem 54, describes lattice-theoretic properties of the maps $\boldsymbol{\rho}$, $\boldsymbol{\tau}$ and $\boldsymbol{\sigma}$. Items (i), (ii) and (iv) in it were first proved in [100]; (iii) is known as the Blok–Esakia theorem [9, 37].

THEOREM 56. (i) *The map $\boldsymbol{\rho}$ is a homomorphism of the lattice* NExt **S4** *onto the lattice* Ext **Int**.

(ii) *The map $\boldsymbol{\tau}$ is an isomorphism from* Ext **Int** *into* NExt **S4**.

(iii) *The map $\boldsymbol{\sigma}$ is an isomorphism from* Ext **Int** *onto* NExt **Grz**.

(iv) *All these maps preserve infinite sums and intersections of logics.*

The following theorem provides a deductive characterisation of the maps $\boldsymbol{\tau}$ and $\boldsymbol{\sigma}$.

THEOREM 57. *For every si-logic L and every modal canonical formula $\alpha(\mathfrak{F}, \mathfrak{D}, \bot)$ built on a quasi-ordered frame $\mathfrak{F}$,*

(i) $\alpha(\mathfrak{F}, \mathfrak{D}, \bot) \in \boldsymbol{\tau}L$ *iff* $\beta(\boldsymbol{\rho}\mathfrak{F}, \boldsymbol{\rho}\mathfrak{D}, \bot) \in L$;

(ii) $\alpha(\mathfrak{F}, \mathfrak{D}, \bot) \in \boldsymbol{\sigma}L$ *iff either $\mathfrak{F}$ is partially ordered and* $\beta(\mathfrak{F}, \mathfrak{D}, \bot) \in L$ *or $\mathfrak{F}$ contains a proper cluster.*

The theorems above can be used for transferring various properties of modal logics to their si-fragments and back. Some results of that sort are collected in Table 5; for proofs see [24].

9.4 Completeness

In this section we briefly discuss the most important general results concerning completeness of si-logics with respect to various classes of Kripke frames. (As in the modal case, the fmp and decidability of a good many concrete si-logics was proved using various forms of filtration; see, e.g., [46, 116, 136, 47, 39].)

That not all si-logics are complete with respect to Kripke frames was discovered by Shehtman [132], who found a way to adjust Fine's [43] idea to the intuitionistic case. As to general positive results, notice first that the preservation theorems yield the following translation of Theorems 24–25 (si-logics of finite width were studied by Sobolev [138]).

THEOREM 58. (1) *Every si-logic of finite depth has the fmp* (*in fact, is locally tabular*).

(2) *Every si-logic of width n is characterised by a class of Noetherian Kripke frames of width $\leq n$.*

PROBLEM 18. Are finitely axiomatisable si-logics of width $n > 1$ decidable? What is their computational complexity?

An intuitionistic formula is said to be *essentially negative* if every occurrence of a variable in it is in the scope of some $\neg$. If φ is essentially negative then $T(\varphi)$ is a $\Box\Diamond$-formula, which—together with Theorem 14—yields the following result of McKay [105]:

THEOREM 59 (McKay). *If a si-logic L is decidable* (*or has the fmp*) *and φ is an essentially negative formula then $L + \varphi$ is decidable* (*has the fmp*).

Say that an occurrence of a variable in a formula is *essential* if it is not in the scope of any $\neg$. A formula φ is *mild* if every two essential occurrences of the same variable in φ are either both positive or both negative. Kuznetsov [83] claimed (we have not seen the proof) that all si-logics whose extra axioms do not contain negative occurrences of essential variables have the fmp. And Wroński [168] announced that if L is a decidable si-logic and φ a mild formula then $L + \varphi$ is also decidable.

Since frames for **Int** contain no clusters, Theorem 44 and its analogue for cofinal subframe logics reduce in the intuitionistic case to the following result which is due to [28, 121, 134, 174]:

THEOREM 60. *All si-logics with disjunction free axioms are first-order definable* (*definable by $\forall\exists$-sentences*) *and $\mathcal{D}$-persistent.*

Nishimura [115] described all si-logics axiomatisable by one-variable formulas. As a consequence of his result and Theorem 19 we obtain the following theorem due to Sobolev [138]:

THEOREM 61 (Sobolev). *All si-logics with extra axioms in one variable have the fmp and are decidable.*

In fact, Sobolev proved a more general (but rather complicated) syntactical sufficient condition of the fmp and constructed a formula in two variables axiomatising a si-logic without the fmp (Shehtman's [132] incomplete si-logic has also axioms in two variables).

By the Blok–Esakia and preservation theorems, the situation with tabular logics in Ext**Int** is the same as in NExt **Grz**. In particular, there are only three pretabular logics in Ext**Int** [95], and the tabularity problem is decidable in Ext**Int**.

9.5 Medvedev's logic

Although this chapter's main concern is *classes* of logics rather than individual systems, we conclude it with a brief discussion of a very elegant and interesting si-logic introduced by Medvedev [109, 110, 111] and known as the *logic of finite problems* or the *Medvedev logic* **ML**. Semantically it can be defined as the set of intuitionistic formulas that are

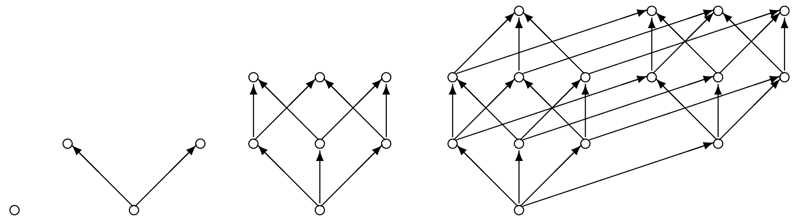


Figure 5.

valid in all 'topless' n-ary Boolean cubes depicted in Fig. 5 for $n = 1, 2, 3, 4$. More precisely, let W_n be the family of nonempty subsets of a set with $n > 0$ elements and xR_ny means $y \subseteq x$, for $x, y \in W_n$. Then

$$\mathbf{ML} \quad = \quad \{\varphi \mid \langle W_n, R_n\rangle \models \varphi \ \text{ for all } n \geq 1\}.$$

It turns out that **ML** is a *constructive* si-logic in the sense that it enjoys the following *disjunction property*

$$\varphi \vee \psi \in \mathbf{ML} \quad \text{implies} \quad \varphi \in \mathbf{ML} \ \text{ or } \ \psi \in \mathbf{ML}$$

and, moreover, no proper extension of **ML** is constructive in this sense [89, 99]. In fact, there are a continuum of maximal constructive si-logics [73, 98, 20, 39], and not a single one of them is known to be finitely axiomatisable or decidable. In particular, [101] shows that **ML** is not finitely axiomatisable.

PROBLEM 19. Does there exist a decidable maximal si-logic with the disjunction property? In particular, is **ML** decidable?

PROBLEM 20. Does there exist a finitely axiomatisable maximal si-logic with the disjunction property?

ACKNOWLEDGEMENTS

We are grateful to our friends and colleagues Aleksander Chagrov and Ian Hodkinson for providing some material for this chapter. Thanks are also due to R. Iemhoff and V. Shehtman for their constructive comments and suggestions. The work on this chapter was partially supported by the U.K. EPSRC grants GR/S61966/01, GR/S63182/01, GR/S63175/01, GR/S61973/01.

BIBLIOGRAPHY

[1] F. Bellissima. On the lattice of extensions of the modal logic $K.Alt_n$. *Archive of Mathematical Logic*, 27:107–114, 1988.

[2] G. Bezhanishvili and S. Ghilardi. An algebraic approach to subframe logics. Part I. *Annals of Pure and Applied Logic* (in print), 2006.

[3] N. Bezhanishvili. Varieties of two-dimensional cylindric algebras. Part I: diagonal-free case. *Algebra Universalis*, 48:11–42, 2002.

[4] N. Bezhanishvili. Varieties of two-dimensional cylindric algebras II. *Algebra Universalis*, 51:177–206, 2004.

[5] N. Bezhanishvili and I. Hodkinson. All normal extensions of $S5$-squared are finitely axiomatizable. *Studia Logica*, 78:443–457, 2004.

[6] N. Bezhanishvili and M. Marx. All proper normal extensions of $S5$-square have the polynomial size model property. *Studia Logica*, 73:367–382, 2003.

[7] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[8] W. Blok. On the degree of incompleteness in modal logics and the covering relation in the lattice of modal logics. Technical Report 78-07, Department of Mathematics, University of Amsterdam, 1978.

[9] W. Blok. *Varieties of interior algebras*. PhD thesis, University of Amsterdam, 1976.

[10] W. Blok. Pretabular varieties of modal algebras. *Studia Logica*, 39:101–124, 1980.

[11] E. Börger, E. Grädel, and Yu. Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997.

[12] L.E.J. Brouwer. *Over de Grondslagen der Wiskunde*. PhD thesis, Amsterdam, 1907. Translation: "On the foundation of mathematics" in A. Heyting, editor, *Brouwer, Collected Works I*, pages 11–101. Elsevier, North-Holland, 1975.

[13] L.E.J. Brouwer. De onbetrouwbaarheid der logische principes. *Tijdschrift voor Wijsbegeerte*, 2:152–158, 1908. Translation: "The unreliability of the logical principles," ibid, pages 107–111.

[14] R.A. Bull. That all normal extensions of $S4.3$ have the finite model property. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 12:341–344, 1966.

[15] R.A. Bull. An algebraic study of tense logic with linear time. *Journal of Symbolic Logic*, 33:27–38, 1968.

[16] A. Chagrov. The lower bound for the cardinality of approximating Kripke frames. In M.I. Kanovich, editor, *Logical Methods for Constructing Effective Algorithms*, pages 96–125. Kalinin State University, Kalinin, 1986. (Russian).

[17] A. Chagrov. Nontabularity—pretabularity, antitabularity, co-antitabularity. In *Algebraic and Logical Constructions*, pages 105–111. Kalinin State University, Kalinin, 1989. (Russian).

[18] A. Chagrov. Undecidable properties of extensions of provability logic. I. *Algebra and Logic*, 29:231–243, 1990.

[19] A. Chagrov. Undecidable properties of extensions of provability logic. II. *Algebra and Logic*, 29:406–413, 1990.

[20] A. Chagrov. Continuality of the set of maximal superintuitionistic logics with the disjunction property. *Mathematical Notes*, 51:188–193, 1992.

[21] A. Chagrov. Undecidable properties of superintuitionistic logics. In S.V. Jablonskij, editor, *Mathematical Problems of Cybernetics, vol.5*, pages 67–108. Physmatlit, Moscow, 1994. (Russian).

[22] A. Chagrov. The algorithmic problem of axiomatising a tablular normal modal logic. In *Logical Investigations*, pages 251–263. Nauka, Moscow, 2002. (Russian).

[23] A. Chagrov and V. Shehtman. Algorithmic aspects of propositional tense logics. Volume 933 of *Lecture Notes in Computer Science*, pages 442–455. Springer, 1995.

[24] A. Chagrov and M. Zakharyaschev. *Modal Logic*, volume 35 of *Oxford Logic Guides*. Clarendon Press, Oxford, 1997.

[25] A. Chagrov and M. Zakharyaschev. The undecidability of the disjunction property of propositional logics and other related problems. *Journal of Symbolic Logic*, 58:49–82, 1993.

[26] A. Chagrov and M. Zakharyaschev. Sahlqvist formulas are not so elementary even above $S4$. In L. Csirmaz, D.M. Gabbay, and M. de Rijke, editors, *Logic Colloquium'92*, pages 61–73. CSLI Publications, Stanford, 1995.

[27] L. Chagrova. On the degree of neighbourhood incompleteness of normal modal logics. In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic 1*, pages 63–72. CSLI Publications, 1998.

[28] L. Chagrova. On the first order definability of intuitionistic formulas with restrictions on occurrences of the connectives. In M.I. Kanovich, editor, *Logical Methods for Constructing Effective Algorithms*, pages 135–136. Kalinin State University, Kalinin, 1986. (Russian).

[29] L. Chagrova. An undecidable problem in correspondence theory. *Journal of Symbolic Logic*, 56:1261–1272, 1991.

[30] M. de Rijke and Y. Venema. Sahlqvist's theorem for Boolean algebras with operators. *Studia Logica*, 54:61–78, 1995.

[31] A. Diego. *Sur les algèbres de Hilbert.* Gauthier-Villars, Paris, 1966.

[32] M. Dummett. A propositional calculus with denumerable matrix. *Journal of Symbolic Logic*, 24:97–106, 1959.

[33] M. Dummett and E. Lemmon. Modal logics between $S4$ and $S5$. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 5:250–264, 1959.

[34] H. Enderton. *A Mathematical Introduction to Logic.* Academic Press, New York, 2nd edition, 2001.

[35] L.L. Esakia. Topological Kripke models. *Soviet Mathematics Doklady*, 15:147–151, 1974.

[36] L.L. Esakia. On varieties of Grzegorczyk algebras. In A. I. Mikhailov, editor, *Studies in Non-classical Logics and Set Theory*, pages 257–287. Moscow, Nauka, 1979. (Russian).

[37] L.L. Esakia. To the theory of modal and superintuitionistic systems. In V.A. Smirnov, editor, *Logical Inference. Proceedings of the USSR Symposium on the Theory of Logical Inference*, pages 147–172. Nauka, Moscow, 1979. (Russian).

[38] L.L. Esakia and V.Yu. Meskhi. Five critical systems. *Theoria*, 40:52–60, 1977.

[39] M. Ferrari and P. Miglioli. Counting the maximal intermediate constructive logics. *Journal of Symbolic Logic*, 58:1365–1408, 1993.

[40] K. Fine. The logics containing $S4.3$. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 17:371–376, 1971.

[41] K. Fine. An ascending chain of $S4$ logics. *Theoria*, 40:110–116, 1974.

[42] K. Fine. An incomplete logic containing $S4$. *Theoria*, 40:23–29, 1974.

[43] K. Fine. Logics containing $K4$, part I. *Journal of Symbolic Logic*, 39:229–237, 1974.

[44] K. Fine. Normal forms in modal logic. *Notre Dame Journal of Formal Logic*, 16:31–42, 1975.

[45] K. Fine. Logics containing $K4$, part II. *Journal of Symbolic Logic*, 50:619–651, 1985.

[46] D. Gabbay. The decidability of the Kreisel–Putnam system. *Journal of Symbolic Logic*, 35:431–436, 1970.

[47] D. Gabbay and D. de Jongh. A sequence of decidable finitely axiomatizable intermediate logics with the disjunction property. *Journal of Symbolic Logic*, 39:67–78, 1974.

[48] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*, volume 148 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2003.

[49] D. Gabelaia, A. Kurucz, F. Wolter, and M. Zakharyaschev. Products of 'transitive' modal logics. *Journal of Symbolic Logic*, 70:993–1021, 2005.

[50] M. Gehrke, H. Nagahashi, and Y. Venema. A Sahlqvist theorem for distributive modal logic. *Annals of Pure and Applied Logic*, 131:65–102, 2005.

[51] K. Gödel. Zum intuitionistischen Aussagenkalkül. *Anzeiger der Akademie der Wissenschaften in Wien*, 69:65–66, 1932.

[52] K. Gödel. Eine Interpretation des intuitionistischen Aussagenkalküls. *Ergebnisse eines mathematischen Kolloquiums*, 4:39–40, 1933.

[53] R. Goldblatt. First-order definability in modal logic. *Journal of Symbolic Logic*, 40:35–40, 1975.

[54] R. Goldblatt. Metamathematics of modal logic, Part I. *Reports on Mathematical Logic*, 6:41–78, 1976.

[55] R. Goldblatt. The McKinsey axiom is not canonical. *Journal of Symbolic Logic*, 56:554–562, 1991.

[56] R. Goldblatt. *Mathematics of Modality.* Number 43 in CSLI Lecture Notes. CSLI Publications, Stanford, 1993.

[57] R. Goldblatt. Mathematical modal logic: a view of its evolution. *Journal of Applied Logic*, 1:309–392, 2003.

[58] R. Goldblatt, I. Hodkinson, and Y. Venema. Erdös graphs resolve Fine's canonicity problem. *Bulletin of Symbolic Logic*, 10:186–208, 2004.

[59] V. Goranko and S. Passy. Using the universal modality: gains and questions. *Journal of Logic and Computation*, 2:5–30, 1992.

[60] V. Goranko and D. Vakarelov. Sahlqvist formulas unleashed in polyadic modal languages. In F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev, editors, *Advances in Modal Logic. Volume 3*, pages 221–240. World Scientific, 2002.

[61] C. Grefe. Modale Logiken funktionaler Frames. Master's thesis, Department of Mathematics, Freie Universität Berlin, 1994.

[62] D. Harel. Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness. *Journal of the ACM*, 33:224–248, 1986.

[63] A. Heyting. Die formalen Regeln der intuitionistischen Logik. *Sitzungsberichte der Preussischen Akademie der Wissenschaften*, pages 42–56, 1930.

[64] I. Hodkinson. Hybrid formulas and elementarily generated modal logics. 2005. Preprint, available at `www.doc.ic.ac.uk/~imh/papers/ns.pdf`.

[65] G.E. Hughes and M.J. Cresswell. *A Companion to Modal Logic.* Methuen, London, 1984.

[66] V.A. Jankov. The relationship between deducibility in the intuitionistic propositional calculus and finite implicational structures. *Soviet Mathematics Doklady*, 4:1203–1204, 1963.

[67] V.A. Jankov. Some superconstructive propositional calculi. *Soviet Mathematics Doklady*, 4:1103–1105, 1963.

[68] V.A. Jankov. The construction of a sequence of strongly independent superintuitionistic propositional calculi. *Soviet Mathematics Doklady*, 9:806–807, 1968.

[69] V.A. Jankov. Conjunctively indecomposable formulas in propositional calculi. *Mathematics of the USSR, Izvestiya*, 3:17–35, 1969.

[70] B. Jónsson. On the canonicity of Sahlqvist identities. *Studia Logica*, 53:473–491, 1994.

[71] B. Jónsson and A. Tarski. Boolean algebras with operators. I. *American Journal of Mathematics*, 73:891–939, 1951.

[72] S. Kikot. A new generalization of Sahlqvist theorem. In *Algebraic and Topological Methods in Non-Classical Logics II*, pages 47–48, Barcelona, 2005.

[73] R.E. Kirk. A result on propositional logics having the disjunction property. *Notre Dame Journal of Formal Logic*, 23:71–74, 1982.

[74] S. Kleene. On the interpretation of intuitionistic number theory. *Journal of Symbolic Logic*, 10:109–124, 1945.

[75] M. Kracht. Modal logics that need large frames. *Notre Dame Journal of Formal Logic*, 40:141–173, 1999.

[76] M. Kracht. *Tools and Techniques in Modal Logic*. Studies in Logic. Elsevier, North-Holland, 1999.

[77] M. Kracht and F. Wolter. Normal monomodal logics can simulate all others. *Journal of Symbolic Logic*, 64:99–138, 1999.

[78] G. Kreisel and H. Putnam. Eine Unableitbarkeitsbeweismethode für den intuitionistischen Aussagenkalkül. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 3:74–78, 1957.

[79] S.A. Kripke. Semantical analysis of modal logic, Part I. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.

[80] J. B. Kruskal. Well-quasi-ordering, the tree theorem and Vazsonyi's conjecture. *Transactions of the American Mathematical Society*, 95:210–225, 1960.

[81] A.V. Kuznetsov. Undecidability of general problems of completeness, decidability and equivalence for propositional calculi. *Algebra and Logic*, 2:47–66, 1963. (Russian).

[82] A.V. Kuznetsov. Some properties of the structure of varieties of pseudo-Boolean algebras. In *Proceedings of the XIth USSR Algebraic Colloquium*, pages 255–256, Kishinev, 1971. (Russian).

[83] A.V. Kuznetsov. The decidability of certain superintuitionistic calculi. In *Proceedings of the IInd USSR Conference on Mathematical Logic*, Moscow, 1972. (Russian).

[84] A.V. Kuznetsov. On superintuitionistic logics. In *Proceedings of the International Congress of Mathematicians*, pages 243–249, Vancouver, 1975.

[85] A.V. Kuznetsov and V.Ya. Gerchiu. Superintuitionistic logics and the finite approximability. *Soviet Mathematics Doklady*, 11:1614–1619, 1970.

[86] E.J. Lemmon. Algebraic semantics for modal logic. I. *Journal of Symbolic Logic*, 31:46–65, 1966.

[87] E.J. Lemmon. Algebraic semantics for modal logic. II. *Journal of Symbolic Logic*, 31:191–218, 1966.

[88] E.J. Lemmon and D.S. Scott. *An Introduction to Modal Logic*. Oxford, Blackwell, 1977.

[89] V.A. Levin. Some syntactic theorems on the calculus of finite problems of Yu.T. Medvedev. *Soviet Mathematics Doklady*, 10:288–290, 1969.

[90] C. Lewis. *A Survey of Symbolic Logic*. University of California Press, Berkeley, 1918.

[91] D. Lewis. Intensional logics without iterative axioms. *Journal of Philosophical logic*, 3:457–466, 1974.

[92] T. Litak. *An algebraic approach to incompleteness in modal logic*. PhD thesis, JAIST, 2005.

[93] T. Litak and F. Wolter. All finitely axiomatizable tense logics of linear time flows are coNP-complete. *Studia Logica*, 81:153–165, 2005.

[94] D. Makinson. Some embedding theorems for modal logic. *Notre Dame Journal of Formal Logic*, 12:252–254, 1971.

[95] L. Maksimova. Pretabular superintuitionistic logics. *Algebra and Logic*, 11:308–314, 1972.

[96] L. Maksimova. Pretabular extensions of Lewis $S4$. *Algebra and Logic*, 14:16–33, 1975.

[97] L. Maksimova. Interpolation theorems in modal logics. Sufficient conditions. *Algebra and Logic*, 19:120–132, 1980.

[98] L. Maksimova. On the number of maximal intermediate logics having the disjunction property. In *Proceedings of the 7th USSR Conference for Mathematical Logic*, page 95. Institute of Mathematics, Novosibirsk, 1984. (Russian).

[99] L. Maksimova. On maximal intermediate logics with the disjunction property. *Studia Logica*, 45:69–75, 1986.

[100] L. Maksimova and V. Rybakov. Lattices of modal logics. *Algebra and Logic*, 13:105–122, 1974.

[101] L. Maksimova, V. Shehtman, and D. Skvortsov. The impossibility of a finite axiomatization of Medvedev's logic of finitary problems. *Soviet Mathematics Doklady*, 20:394–398, 1979.

[102] A. Mal'cev. *Algorithms and Recursive Functions*. Wolters-Noordhoff, Groningen, 1970.

[103] M. Marx. Complexity of products of modal logics. *Journal of Logic and Computation*, 9:197–214, 1999.

[104] C. McKay. The decidability of certain intermediate logics. *Journal of Symbolic Logic*, 33:258–264, 1968.

[105] C. McKay. A class of decidable intermediate propositional logics. *Journal of Symbolic Logic*, 36:127–128, 1971.

[106] R. McKenzie. Equational bases and non-modular lattice varieties. *Transactions of the American Mathematical Society*, 174:1–43, 1972.

[107] J.C.C. McKinsey and A. Tarski. On closed elements in closure algebras. *Annals of Mathematics*, 47:122–162, 1946.

[108] J.C.C. McKinsey and A. Tarski. Some theorems about the sentential calculi of Lewis and Heyting. *Journal of Symbolic Logic*, 13:1–15, 1948.

[109] Yu. Medvedev. Finite problems. *Soviet Mathematics Doklady*, 3:227–230, 1962.

[110] Yu. Medvedev. Interpretation of logical formulas by means of finite problems and its relation to the realizability theory. *Soviet Mathematics Doklady*, 4:180–183, 1963.

[111] Yu. Medvedev. Interpretation of logical formulas by means of finite problems. *Soviet Mathematics Doklady*, 7:857–860, 1966.

[112] F. Montagna. Provability in finite subtheories of PA and relative interpretability: a modal investigation. *Journal of Symbolic Logic*, 52:494–511, 1987.

[113] M.C. Nagle. The decidability of normal $K5$-logics. *Journal of Symbolic Logic*, 46:319–328, 1981.

[114] M.C. Nagle and S.K. Thomason. The extensions of the modal logic $K5$. *Journal of Symbolic Logic*, 50:102–108, 1985.

[115] I. Nishimura. On formulas of one variable in intuitionistic propositional calculus. *Journal of Symbolic Logic*, 25:327–331, 1960.

[116] H. Ono. Some results on the intermediate logics. *Publications of the Research Institute for Mathematical Science, Kyoto University*, 8:117–130, 1972.

[117] I. Orlov. The calculus of compatibility of propositions. *Mathematics of the USSR, Sbornik*, 35:263–286, 1928. (In Russian).

[118] H. Rasiowa and R. Sikorski. *The Mathematics of Metamathematics*. Polish Scientific Publishers, 1963.

[119] W. Rautenberg. *Klassische und nichtklassische Aussagenlogik*. Vieweg, Braunschweig-Wiesbaden, 1979.

[120] M. Reynolds and M. Zakharyaschev. On the products of linear modal logics. *Journal of Logic and Computation*, 11:909–931, 2001.

[121] P. Rodenburg. *Intuitionistic correspondence theory*. PhD thesis, University of Amsterdam, 1986.

[122] G. Rose. Propositional calculus and realizability. *Transactions of the American Mathematical Society*, 75:1–19, 1953.

[123] V.V. Rybakov. Noncompact extensions of the logic $S4$. *Algebra and Logic*, 16:321–334, 1977.

[124] V.V. Rybakov. Modal logics with LM-axioms. *Algebra and Logic*, 17:302–310, 1978.

[125] H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logic. In S. Kanger, editor, *Proceedings of the Third Scandinavian Logic Symposium*, pages 110–143. North-Holland, 1975.

[126] G. Sambin and V. Vaccaro. A topological proof of Sahlqvist's theorem. *Journal of Symbolic Logic*, 54:992–999, 1989.

[127] S.J. Scroggs. Extensions of the Lewis system $S5$. *Journal of Symbolic Logic*, 16:112–120, 1951.

[128] K. Segerberg. Modal logics with linear alternative relations. *Theoria*, 36:301–322, 1970.

[129] K. Segerberg. An essay in classical modal logic. *Philosophical Studies*, 13, 1971.

[130] K. Segerberg. Two-dimensional modal logic. *Journal of Philosophical Logic*, 2:77–96, 1973.

[131] K. Segerberg. Modal logics with functional alternative relations. *Notre Dame Journal of Formal Logic*, 27:504–522, 1986.

[132] V. Shehtman. On incomplete propositional logics. *Soviet Mathematics Doklady*, 18:985–989, 1977.

[133] V. Shehtman. Topological models of propositional logics. *Semiotics and Information Science*, 15:74–98, 1980. (Russian).

[134] T. Shimura. Kripke completeness of some intermediate predicate logics with the axiom of constant domain and a variant of canonical formulas. *Studia Logica*, 52:23–40, 1993.

[135] A. Sistla and E. Clarke. The complexity of propositional linear temporal logics. *Journal of the Association for Computing Machinery*, 32:733–749, 1985.

[136] C. Smoryński. *Investigations of Intuitionistic Formal Systems by means of Kripke Frames*. PhD thesis, University of Illinois, 1973.

[137] C. Smoryński. *Self-reference and Modal Logic*. Springer Verlag, Heidelberg & New York, 1985.

[138] S.K. Sobolev. On the finite approximability of superintuitionistic logics. *Mathematics of the USSR, Sbornik*, 31:257–268, 1977.

[139] E. Spaan. *Complexity of Modal Logics*. PhD thesis, Department of Mathematics and Computer Science, University of Amsterdam, 1993.
[140] S. Thomason. Noncompactness in propositional modal logic. *Journal of Symbolic Logic*, 37:716–720, 1972.
[141] S. Thomason. Semantic analysis of tense logic. *Journal of Symbolic Logic*, 37:150–158, 1972.
[142] S. Thomason. An incompleteness theorem in modal logic. *Theoria*, 40:30–34, 1974.
[143] S. Thomason. Reduction of tense logic to modal logic I. *Journal of Symbolic Logic*, 39:549–551, 1974.
[144] S. Thomason. The logical consequence relation of propositional tense logic. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 21:29–40, 1975.
[145] S. Thomason. Reduction of second-order logic to modal logic. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 21:107–114, 1975.
[146] S. Thomason. Undecidability of the completeness problem of modal logic. In *Universal Algebra and Applications, Banach Center Publications, vol.9*, pages 341–345, Warsaw, 1982. PNW–Polish Scientific Publishers.
[147] E. Tomaszewski. Algebraic approach to canonical formulas in ExtFL_{ew} and NExt K. In *Algebraic and Topological Methods in Non-Classical Logics II*, pages 82–83, Barcelona, 2005.
[148] G.S. Tseitin. Associative calculus with unsolvable equivalence problem. *Proceedings of the Mathematical Steklov Institute of the USSR Academy of Sciences*, 52:172–189, 1958. Translation: American Mathematical Society. Translations. Series 2. 94:73–92.
[149] T. Umezawa. Über die Zwischensysteme der Aussagenlogik. *Nagoya Mathematical Journal*, 9:181–189, 1955.
[150] T. Umezawa. On intermediate propositional logics. *Journal of Symbolic Logic*, 24:20–36, 1959.
[151] A. Urquhart. Implicational formulas in intuitionistic logic. *Journal of Symbolic Logic*, 39:661–664, 1974.
[152] J. van Benthem. Some kinds of modal completeness. *Studia Logica*, 39:125–141, 1980.
[153] J. van Benthem. Correspondence theory. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, vol.2*, pages 167–247. Reidel, Dordrecht, 1984.
[154] J. van Benthem. A note on modal formulas and relational properties. *Journal of Symbolic Logic*, 40:85–88, 1975.
[155] J. van Benthem. Modal reduction principles. *Journal of Symbolic Logic*, 41:301–312, 1976.
[156] J. van Benthem. Canonical modal logics and ultrafilter extensions. *Journal of Symbolic Logic*, 44:1–8, 1979.
[157] D. van Dalen. Intuitionistic Logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, vol.3*, pages 225–339. Reidel, Dordrecht, 1986.
[158] M. Vardi. Why is modal logic so robustly decidable? In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol.31*, pages 149–184. AMS, 1997.
[159] A. Visser. A course in bimodal provability logic. *Annals of Pure and Applied Logic*, 73:115–142, 1995.
[160] F. Wolter. The finite model property in tense logic. *Journal of Symbolic Logic*, 60:757–774, 1995.
[161] F. Wolter. A counterexample in tense logic. *Notre Dame Journal of Formal Logic*, 37:167–173, 1996.
[162] F. Wolter. Properties of tense logics. *Mathematical Logic Quarterly*, 42:481–500, 1996.
[163] F. Wolter. Tense logics without tense operators. *Mathematical Logic Quarterly*, 42:145–171, 1996.
[164] F. Wolter. Completeness and decidability of tense logics closely related to logics containing $K4$. *Journal of Symbolic Logic*, 62:131–158, 1997.
[165] F. Wolter. A note on the interpolation property in tense logic. *Journal of Philosophical Logic*, 26:545–551, 1997.
[166] F. Wolter. The structure of lattices of subframe logics. *Annals of Pure and Applied Logic*, 86:47–100, 1997.
[167] F. Wolter. All finitely axiomatizable subframe logics containing CSM are decidable. *Archive for Mathematical Logic*, 37:167–182, 1998.
[168] A. Wroński. Sufficient condition of decidability for intermediate propositional logics. In *ASL Logic Colloquium, Berlin'89*, 1989.
[169] M. Zakharyaschev. On intermediate logics. *Soviet Mathematics Doklady*, 27:274–277, 1983.
[170] M. Zakharyaschev. Syntax and semantics of intermediate logics. *Algebra and Logic*, 28:262–282, 1989.
[171] M. Zakharyaschev. Modal companions of superintuitionistic logics: syntax, semantics and preservation theorems. *Mathematics of the USSR, Sbornik*, 68:277–289, 1991.
[172] M. Zakharyaschev. Canonical formulas for $K4$. Part I: Basic results. *Journal of Symbolic Logic*, 57:1377–1402, 1992.
[173] M. Zakharyaschev. A sufficient condition for the finite model property of modal logics above $K4$. *Bulletin of the IGPL*, 1:13–21, 1993.
[174] M. Zakharyaschev. Canonical formulas for $K4$. Part II: Cofinal subframe logics. *Journal of Symbolic Logic*, 61:421–449, 1996.
[175] M. Zakharyaschev. Canonical formulas for $K4$. Part III: The finite model property. *Journal of Symbolic Logic*, 62:950–975, 1997.

[176] M. Zakharyaschev and A. Alekseev. All finitely axiomatizable normal extensions of $K4.3$ are decidable. *Mathematical Logic Quarterly*, 41:15–23, 1995.
[177] M. Zakharyaschev, F. Wolter, and A. Chagrov. Advanced modal logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, vol.3*, pages 83–266. Kluwer Academic Publishers, 2nd edition, 2001.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

8

MODAL CONSEQUENCE RELATIONS

Marcus Kracht

1 Introduction . . . 492
2 Basic Theory of Modal Consequence Relations . . . 493
2.1 Consequence Relations . . . 494
2.2 Rules . . . 496
2.3 The Deduction Theorem . . . 497
2.4 Interpolation . . . 498
2.5 Modal Logics and Modal Consequence Relations . . . 499
2.6 Lattices of Modal Consequence Relations . . . 500
2.7 The Locale of Modal Logics—General Theory . . . 504
2.8 Splittings . . . 505
2.9 Some Splittings . . . 507
2.10 Axiomatisation Bases . . . 508
3 The Local and the Global . . . 510
3.1 Equivalential and Algebraisable Logics . . . 510
3.2 Global Consequence Relations and Logics . . . 513
3.3 Semisimple Varieties of Modal Algebras . . . 514
4 Reduction to Monomodal Logic . . . 516
4.1 Simulating Two Operators by One . . . 516
4.2 Algebraic Properties of the Simulation . . . 518
4.3 Unsimulation . . . 519
4.4 The Main Theorem . . . 521
4.5 Simulating Polyadic and Nonstandard Operators by Monadic Operators . . . 522
5 Interpolation . . . 525
5.1 Algebraic Characterisation . . . 525
5.2 Proving Interpolation . . . 528
5.3 Beth Properties . . . 531
5.4 Fixed Point Theorems . . . 532
5.5 Uniform Interpolation . . . 534
6 Admissible Rules . . . 536
6.1 General Theory . . . 536
6.2 Frame Characterisation of Admissibility . . . 538
6.3 Axiomatizing the Admissible rules . . . 540
6.4 Decidability of the Admissibility of a Rule . . . 541
6.5 Structural Completeness . . . 541
7 Further Topics . . . 542

1 INTRODUCTION

Logic is generally defined as the science of reasoning. Mathematical logic is mainly concerned with forms of reasoning that lead from true premises to true conclusions. Thus we say that the argument from $\sigma_0; \sigma_1; \cdots ; \sigma_{n-1}$ to δ is *logically correct* if whenever σ_i is true for all $i < n$, then so is δ. In place of 'argument' one also speaks of 'inference'. The language object '$\sigma_0; \sigma_1; \cdots ; \sigma_{n-1}/\delta$' is is called a *rule*, of which arguments are instances. A rule is *valid* if all its instances are. Central to this approach is the notion of a *consequence relation*, which is a relation between sets of formulae and formulae. A consequence relation $\vdash$ specifies which arguments are valid; the argument from a set Σ to a formula δ is valid in $\vdash$ iff $\langle \Sigma, \delta \rangle \in \vdash$, for which we write $\Sigma \vdash \delta$. δ is a *tautology* of $\vdash$ if $\varnothing \vdash \delta$, for which we also write $\vdash \delta$.

In the early years, research into modal logic was concerned with the question of finding the correct inference rules. This research line is still there but has been marginalized by the research into modal *logics*, where a logic is just a set of formulae; this set is the set of tautologies of a certain consequence relation, but many consequence relations share the same tautologies. The shift of focus in the research has to do in part with the precedent set by predicate logic: predicate logic is standardly axiomatized in a Hilbert-style fashion, which fixes the inference rules and leaves only the axioms as a parameter. Another source may have been the fact that there is a biunique correspondence between varieties of modal algebras and axiomatic extensions of K, which allowed for rather deep investigations into the space of logics, using the machinery of equational theories. This research led to deep results on the structure of the lattice of modal logics and benefits also the research into consequence relations. Recently, however, algebraic logic has provided more and more tools that allow to extend the algebraic method to the study of consequence relations in general (see for example [60] and [14]). In particular the investigations into the Leibniz operator initiated by Blok and Pigozzi in [5] have brought new life into the discussion and allow to see a much broader picture than before.

Now, even if one is comfortable with classical logic, it is not immediately clear what the correct inferences are in modal logic. The first problem is that it is not generally agreed what the meaning of the modal operator(s) is or should be. In fact, rather than a drawback, the availability of very many different interpretations is the strength of modal logic; it gives flexibility, however at the price that there is not one modal logic, there are uncountably many. For example, $\Box$ as metaphysical necessity satisfies **S5**, $\Box$ as provability in **PA** satisfies **G**, $\Box$ as future necessity (arguably) satisfies **S4.3**, and so on. This is in part because the interpretation decides which algebras are suitable (intended) and which ones are not. However, there is another parameter of variation, and this is the notion of truth itself. In the most popular interpretation, truth is truth at a world; but we could also understand it as truth in every world of the structure. The two give rise to two distinct consequence relations, the *local* and the *global*, which very often do not coincide even though they always have the same set of tautologies. If truth is defined to be truth at every world under all substitutions we finally arrive at the maximal consequence relation compatible with a logic, in which a rule is derived iff it is admissible for that logic. It is this plurality of interpretations that gives rise to the different topics of this contribution and provides the underlying thread that connects them.

The paper is organised as follows. We shall first review basic concepts from universal algebra and basic logical notions such as consequence relations, rules, the deduction theorem and interpolation; then we shall briefly look at modal consequence relations and the structure of the lattice they form; finally, we turn to the notion of a splitting. This concludes Section 2. In Section 3 we shall look at local and global consequence relations. The first part will deal with consequence relations from an algebraic perspective; the second part studies global consequence relations in more detail and the third part outlines the connection between semisimple varieties of modal algebras and weak transitivity. The next section deals with reductions of polymodal and polyadic modal logics to monomodal logic. It reviews results that establish that the lattices of polymodal and polyadic logics can be naturally embedded into the lattice of monomodal logics preserving and reflecting a good deal of properties. This justifies ex post the almost exclusive study of monomodal logics in spite of the practical usefulness of polymodal and polyadic logics. Section 5 looks at interpolation. In detail, it shall give an algebraic characterisation of interpolation and ways of establishing interpolation for logics. Next we shall look at Beth-definability and fixed point theorems and finally at uniform interpolation. Section 6 is devoted to admissible rules. In particular, it deals with questions of axiomatisability of the set of admissible rules, and with the problem of deciding whether a given rule is admissible in a logic. Finally, in Section 7 we take a brief look at more general notions of a rule, like multiple conclusion rules.

2 BASIC THEORY OF MODAL CONSEQUENCE RELATIONS

This chapter makes heavy use of notions from universal algebra. The reader is referred to Chapter 6 for background information concerning universal algebra and in particular the theory of BAOs and how they relate to (general) frames. We shall quickly review some terminology. A **signature** is a pair $\langle F, \nu\rangle$, where F is a set of so-called **function symbols** or **connectives** and $\nu : F \to \omega$ a function assigning to each symbol an arity. **Terms** are expressions of this language based on variables. We shall also refer to ν alone as a signature. We shall assume that the reader is acquainted with basic notions of universal algebra, such as a **ν-algebra**. Given a map $v : X \to A$ from a set X of variables into the underlying set of A, there is at most one homomorphic extension $\overline{v} : \mathfrak{Tm}_\nu(X) \to \mathfrak{A}$, where $\mathfrak{Tm}_\nu(X)$ denotes the algebra of terms in the signature ν over the set X (whose underlying set is $\mathrm{Tm}_\nu(X)$). On a ν-algebra $\mathfrak{A}$, terms induce **term functions** in the obvious way. If we allow to expand the signature by a constant $\underline{a}$ for every $a \in A$, the term functions induced by this enriched language on $\mathfrak{A}$ are called **polynomials**. In what is to follow, terms will also be called **formulae**, F will always contain $\top$, $\wedge$ and $\neg$, and $\nu(\top) = 0$, $\nu(\neg) = 1$ and $\nu(\wedge) = 2$. Moreover, F will additionally contain connectives $\Box_i$, $i < \kappa$, called **modal operators**, which are unary unless otherwise stated. κ need not be finite. The relation corresponding to $\Box_i$ will standardly be denoted by $\lhd_i$. The set of variables is $V := \{p_i : i \in \omega\}$. Sets of formulae are denoted in the usual way using the semicolon notation: $\Delta; \chi$ abbreviates $\Delta \cup \{\chi\}$. We write $\mathrm{var}(\varphi)$ for the set of variables occurring in φ, and $\mathrm{sf}(\varphi)$ for the set of subformulae of φ. Similarly, $\mathrm{var}(\Delta)$ and $\mathrm{sf}(\Delta)$ are used for sets of formulae. A **substitution** is defined by a map $s : V \to \mathrm{Tm}_\nu(V)$. $s(\varphi)$ or φ^s denotes the effect on φ of performing the substitution s.

2.1 Consequence Relations

DEFINITION 1. Let $\mathrm{Tm}_\nu(V)$ be a propositional language. A **consequence relation** over $\mathrm{Tm}_\nu(V)$ is a relation $\vdash \subseteq \wp(\mathrm{Tm}_\nu(V)) \times \mathrm{Tm}_\nu(V)$ between sets of formulae and a single formula such that

1. $\varphi \vdash \varphi$
2. $\Delta \vdash \varphi$ and $\Delta \subseteq \Delta'$ implies $\Delta' \vdash \varphi$.
3. $\Delta \vdash \chi$ and $\Sigma; \chi \vdash \varphi$ implies $\Delta; \Sigma \vdash \varphi$.

$\vdash$ is **structural** if from $\Delta \vdash \varphi$ follows $\Delta^s \vdash \varphi^s$, where s is a substitution. $\vdash$ is **finitary** (or **compact**) if from $\Delta \vdash \varphi$ follows that there is a finite $\Delta' \subseteq \Delta$ such that $\Delta' \vdash \varphi$. A **tautology** of $\vdash$ is a formula φ such that $\vdash \varphi$. $\mathrm{Taut}(\vdash)$ is the set of tautologies of $\vdash$.

There is an alternative approach via deductively closed sets and via closure operators (see Surma [55] for a discussion of alternatives to consequence relations). Given $\vdash$, let $\Sigma^\vdash := \{\varphi : \Sigma \vdash \varphi\}$. The sets of the form $\Sigma^\vdash$ are called **theories** of $\vdash$. Then the following holds.

1. $\Sigma \subseteq \Sigma^\vdash$.
2. $\Sigma^{\vdash\vdash} \subseteq \Sigma^\vdash$.

$\vdash$ is structural iff for all substitutions s and all Σ

$$\Sigma^{\vdash s} \subseteq \Sigma^{s\vdash} \tag{1}$$

$\vdash$ is finitary iff for all Σ

$$\Sigma^\vdash = \bigcup\{\Sigma_0^\vdash : \Sigma_0 \subseteq \Sigma, \Sigma_0 \text{ finite}\} \tag{2}$$

A characterisation of a finitary structural consequence relation in terms of its theories is as follows.

1. The language is a $\vdash$-theory.
2. Every intersection of $\vdash$-theories is a $\vdash$-theory.
3. If T is a $\vdash$-theory, so is $s^{-1}(T)$.
4. If T_i, $i \in \omega$, is an ascending chain of $\vdash$-theories, $\bigcup T_i$ is a $\vdash$-theory.

For the general theory of consequence relation see [60]. For consequence relations and modal logic see [50]. In the sequel, unless otherwise stated, consequence relations are assumed to be finitary and structural. The signatures are signatures extending classical propositional logic by some (typically unary) modal operators.

One can think of a finitary consequence relation as a first order theory of formulae in the following way. A statement of the form $\Delta \vdash \varphi$ is rendered

$$(\forall \vec{x})(\bigwedge\langle T(\delta) : \delta \in \Delta\rangle) \to T(\varphi) \tag{3}$$

where T is a newly introduced predicate; the universal quantifier binds off the free variables occurring in all the formulae. Given this interpretation, the appropriate structures to interpret consequence relation in are matrices in the sense of the following definition.

DEFINITION 2. A ν-**matrix** for a signature ν is a pair $\mathfrak{M} = \langle \mathfrak{A}, D\rangle$ where $\mathfrak{A}$ is a ν-algebra and $D \subseteq A$ a subset. A is called the set of **truth values** and D the set of **designated truth values**. An **assignment** or a **valuation** into $\mathfrak{M}$ is a map v from the set of variables into A. v makes φ **true** in $\mathfrak{M}$ if $\overline{v}(\varphi) \in D$; otherwise it makes φ **false**.

Given a matrix $\mathfrak{M}$ we can define a relation $\vdash_{\mathfrak{M}}$ by

$$(4) \qquad \Delta \vdash_{\mathfrak{M}} \varphi \quad \Leftrightarrow \quad \text{for all assignments } v: \text{ If } \overline{v}[\Delta] \subseteq D \text{ then } \overline{v}(\varphi) \in D$$

If $\vdash \subseteq \vdash_{\mathfrak{M}}$ then we also say that $\mathfrak{M}$ is a **matrix for** $\vdash$. Given $\mathfrak{A}$, we say that D is a **filter** for $\vdash$ if D is closed under the rules; equivalently D is a filter if $\vdash_{\langle \mathfrak{A}, D\rangle} \supseteq \vdash$. Given a class $\mathcal{S}$ of matrices (for the same signature) we define

$$(5) \qquad \vdash_{\mathcal{S}} := \bigcap \langle \vdash_{\mathfrak{M}} : \mathfrak{M} \in \mathcal{S}\rangle$$

THEOREM 3. *Let ν be a signature. For each class $\mathcal{S}$ of ν-matrices $\vdash_{\mathcal{S}}$ is a (possibly nonfinitary) consequence relation.*

THEOREM 4 (Wójcicki). *For every structural consequence relation $\vdash$ there exists a class $\mathcal{S}$ of matrices such that $\vdash = \vdash_{\mathcal{S}}$.*

Proof. Given the language, let S consist of all $\langle \mathfrak{Tm}_\nu(V), T\rangle$ where T is a theory of $\vdash$. First we show that for each such matrix $\mathfrak{M}$, $\vdash \subseteq \vdash_{\mathfrak{M}}$. To that end, assume $\Sigma \vdash \varphi$ and that $\overline{v}[\Sigma] \subseteq T$. Now $\overline{v}$ is in fact a substitution, and T is deductively closed, and so $\overline{v}(\varphi) \in T$ as well, as required. Now assume $\Sigma \nvdash \varphi$. We have to find a single matrix $\mathfrak{M}$ of this form such that $\Sigma \nvdash_{\mathfrak{M}} \varphi$. For example, $\mathfrak{M} := \langle \mathfrak{Tm}_\nu(V), \Sigma^{\vdash}\rangle$. Then with $\overline{v}$ the identity map, $\overline{v}[\Sigma] = \Sigma \subseteq \Sigma^{\vdash}$. However, $\overline{v}(\varphi) = \varphi \notin \Sigma^{\vdash}$ by definition of $\Sigma^{\vdash}$ and the fact that $\Sigma \nvdash \varphi$. ❑

If $\mathfrak{M}$ is a matrix for $\vdash$, then the set of truth values must be closed under the rules. The previous theorem can be refined somewhat. Let $\mathfrak{M} = \langle \mathfrak{A}, D\rangle$ be a logical matrix, and Θ a congruence on $\mathfrak{A}$. We write $[x]\Theta := \{y : x \;\Theta\; y\}$. The sets $[x]\Theta$ are called **blocks** of the congruence. Θ is called a **matrix congruence** if D is a union of Θ-blocks, that is, if $x \in D$ then $[x]\Theta \subseteq D$. In that case we can reduce the whole matrix by Θ and define $\mathfrak{M}/\Theta := \langle \mathfrak{A}/\Theta, D/\Theta\rangle$. The following is easy to show.

LEMMA 5. *Let $\mathfrak{M}$ be a matrix and Θ a matrix congruence of $\mathfrak{M}$. Then $\vdash_{\mathfrak{M}} = \vdash_{\mathfrak{M}/\Theta}$.*

Call a matrix **reduced** if the diagonal, that is the relation $\Delta = \{\langle x, x\rangle : x \in A\}$, is the only matrix congruence. We can sharpen Theorem 4 to the following

THEOREM 6. *For each logic $\langle \mathcal{L}, \vdash\rangle$ there exists a class S of reduced matrices such that $\vdash = \vdash_S$.*

Let $\mathcal{S}$ be a class of ν-matrices. $\mathcal{S}$ is called a **unital semantics** for $\vdash$ if $\vdash = \vdash_{\mathcal{S}}$ and for all $\langle \mathfrak{A}, D\rangle \in \mathcal{S}$ we have $|D| \leq 1$. (See Janusz Czelakowski [12, 13]. A unital semantics is often called **algebraic**. This, however, is different from the notion of 'algebraic' discussed

by Wim Blok and Don Pigozzi in [5].) The following is a useful fact, which is not hard to verify.

PROPOSITION 7. *Let* $\vdash$ *have a unital semantics. Then in* $\vdash$ *the rules* $p; q; \varphi(p) \vdash \varphi(q)$ *are valid for all formulae* φ.

Notice that when a logic over a language $\mathcal{L}$ is given and an algebra $\mathfrak{A}$ with appropriate signature, the set of designated truth values must always be a deductively closed set, otherwise the resulting matrix is not a matrix for the logic. A theory is **consistent** if it is not the entire language, and **maximal consistent** if it is maximal in the set of consistent theories. Every consistent theory is contained in a maximally consistent theory. For classical logics the construction in the proof of Theorem 4 can be strengthened by taking as matrices in S those containing only maximally consistent theories. For if $\Sigma \nvdash \varphi$ then $\Sigma; \neg\varphi$ is consistent and so for some maximal consistent Δ containing Σ we have $\neg\varphi \in \Delta$. Taking v to be the identity, $\overline{v}[\Sigma] = \Sigma \subseteq \Delta$, but $\overline{v}(\varphi) \notin \Delta$, otherwise Δ is not consistent.

2.2 Rules

A **rule** is a pair $\rho = \langle \Delta, \varphi \rangle$, where Δ is a set of formulae, and δ a single formula. We also write Δ/δ. If Δ is finite, we call ρ **finitary**; and if Δ is empty, we call ρ an **axiom**. ρ is n-ary if $|\Delta| = n$. ρ is a **derived** rule of $\vdash$ if $\rho \in \vdash$. ρ is **admissible** if for every substitution s: if $\Delta^s \subseteq \mathrm{Taut}(\vdash)$ then $\varphi^s \in \mathrm{Taut}(\vdash)$.

If R is a set of finitary rules, $\vdash^R$ denotes the smallest finitary, structural consequence relation that contains R. Given a consequence relation $\vdash$ and a rule ρ, $\vdash^{+\rho}$ is the least consequence relation containing $\vdash$ and ρ. $\vdash$ is called **consistent** if it is not the maximal relation. $\vdash$ is consistent iff p is not a tautology. For a consistent $\vdash$ put

$$(6) \qquad E(\vdash) := \{n : \text{there is an } n\text{-ary rule } \rho \notin \vdash \text{ such that } \nvdash^{+\rho} p\}$$

$\vdash$ is called **Post-complete** if $0 \notin E(\vdash)$. It is **structurally complete** if every admissible rule is derivable.

PROPOSITION 8 (Tokarz). *(1)* $\vdash$ *is structurally complete iff* $E(\vdash) \subseteq \{0\}$. *(2)* $\vdash$ *is maximal consistent iff it is both structurally complete and Post-complete.*

There is a special matrix, $\mathfrak{Taut} = \langle \mathfrak{Tm}_\nu(V), \varnothing^\vdash \rangle$. Recall that $\varnothing^\vdash$ are simply the tautologies of a logic.

THEOREM 9 (Wójcicki). $\vdash$ *is structurally complete iff* $\vdash = \vdash_{\mathfrak{Taut}}$.

$\vdash^R$ can be described as follows. If s is a substitution, say that $\langle \Delta^s, \varphi^s \rangle$ is an **instance** of $\langle \Delta, \varphi \rangle$. An R-**proof** of φ from Σ is a sequence $\langle \delta_i : i < n+1 \rangle$ such that $\delta_n = \varphi$, and for every $i < n+1$: either $\delta_i \in \Sigma$ or there are $j_k < i$, $k < p$, such that $\langle \{\delta_{j_k} : k < p\}, \delta_i \rangle$ is an instance of a rule from R.

PROPOSITION 10. $\Sigma \vdash^R \varphi$ *iff there exists an* R*-proof of* φ *from* Σ.

We remark here that $\vdash$ is finitary iff there is a set R of finitary rules such that $\vdash = \vdash^R$. Of course, R may be infinite. $\vdash$ is **decidable** if for all finite Σ and all φ we can decide whether or not $\Sigma \vdash \varphi$. The following is from [32].

THEOREM 11 (Harrop). *Suppose that* $\mathfrak{M} = \langle \mathfrak{A}, D \rangle$ *is a finite logical matrix. Then* $\vdash_{\mathfrak{M}}$ *is decidable.*

For example, one can use truth-tables. This procedure is generally slower than tableaux-methods, but only mildly so (see [15]).

2.3 The Deduction Theorem

The rule of **modus ponens** ($\mathrm{MP}_{\twoheadrightarrow}$) for a binary connective $\twoheadrightarrow$ is the rule $\langle\{p, p \twoheadrightarrow q\}, q\rangle$. ($\mathrm{MP}_{\rightarrow}$) is called (MP) in classical logic. There are many more connectives $\twoheadrightarrow$ for which ($\mathrm{MP}_{\twoheadrightarrow}$) is a derived rule, for example $\wedge$. $\twoheadrightarrow$ is said to satisfy a **deduction theorem** with respect to $\vdash$ if for all Σ, φ, ψ

$$(7) \qquad \Sigma; \varphi \vdash \psi \quad \Leftrightarrow \quad \Sigma \vdash \varphi \twoheadrightarrow \psi$$

A consequence relation $\vdash$ is said to **satisfy the deduction theorem** (DT) for $\twoheadrightarrow$ if $\twoheadrightarrow$ satisfies ($\mathrm{MP}_{\twoheadrightarrow}$) and (7) holds. (See [14] for a survey of deduction theorems.) Given (DT) it is possible to transform any rule different from ($\mathrm{MP}_{\twoheadrightarrow}$) into an axiom preserving the consequence relation. Hence it is possible to replace the original rule calculus by a Hilbert-style calculus, where ($\mathrm{MP}_{\twoheadrightarrow}$) is the only rule which is not an axiom. Given a set of rules R, we say it has a **deduction theorem** for $\twoheadrightarrow$ if $\vdash^R$ does.

THEOREM 12. *A Hilbert-style calculus for $\twoheadrightarrow$ has a deduction theorem for $\twoheadrightarrow$ iff $\twoheadrightarrow$ satisfies ($MP_{\twoheadrightarrow}$) and the following are axioms of $\vdash$:*

$$(8) \qquad p \twoheadrightarrow (q \twoheadrightarrow p)$$

$$(9) \qquad (p \twoheadrightarrow (q \twoheadrightarrow r)) \twoheadrightarrow ((p \twoheadrightarrow q) \twoheadrightarrow (p \twoheadrightarrow r))$$

Proof. ($\Rightarrow$) Suppose both ($\mathrm{MP}_{\twoheadrightarrow}$) and (7) hold for $\twoheadrightarrow$. Then, since $\varphi \vdash \varphi$, also $\varphi; \psi \vdash \varphi$ and (by (7)) also $\varphi \vdash \psi \twoheadrightarrow \varphi$ and (again by (7)) $\vdash \varphi \twoheadrightarrow (\psi \twoheadrightarrow \varphi)$. For (9) note that the following sequence

$$(10) \qquad \langle \varphi \twoheadrightarrow (\psi \twoheadrightarrow \chi), \varphi \twoheadrightarrow \psi, \varphi, \psi \twoheadrightarrow \chi, \psi, \chi\rangle$$

proves $\varphi \twoheadrightarrow (\psi \twoheadrightarrow \chi); \varphi \twoheadrightarrow \psi; \varphi \vdash \chi$. Apply (DT) three times and the formula is proved. ($\Leftarrow$) By induction on the length of an R-proof $\vec{\alpha}$ of ψ from $\Sigma \cup \{\varphi\}$ we show that $\Sigma \vdash \varphi \twoheadrightarrow \psi$. Suppose the length of $\vec{\alpha}$ is 1. Then $\psi \in \Sigma \cup \{\varphi\}$. There are two cases: (1) $\psi \in \Sigma$. Then observe that $\langle \psi \twoheadrightarrow (\varphi \twoheadrightarrow \psi), \psi, \varphi \twoheadrightarrow \psi\rangle$ is a proof of $\varphi \twoheadrightarrow \psi$ from Σ. (2) $\psi = \varphi$. Then we have to show that $\Sigma \vdash \varphi \twoheadrightarrow \varphi$. Now observe that the following is an instance of (9):

$$(11) \qquad (\varphi \twoheadrightarrow ((\psi \twoheadrightarrow \varphi) \twoheadrightarrow \varphi)) \twoheadrightarrow ((\varphi \twoheadrightarrow (\psi \twoheadrightarrow \varphi)) \twoheadrightarrow (\varphi \twoheadrightarrow \varphi))$$

But $\varphi \twoheadrightarrow ((\psi \twoheadrightarrow \varphi) \twoheadrightarrow \varphi)$ and $\varphi \twoheadrightarrow (\psi \twoheadrightarrow \varphi)$ are both instances of (8) and by applying ($\mathrm{MP}_{\twoheadrightarrow}$) twice we get $\varphi \twoheadrightarrow \varphi$. Now let $\vec{\alpha}$ be of length > 1. Then we may assume that ψ is obtained by an application of ($\mathrm{MP}_{\twoheadrightarrow}$) from some formulae χ and $\chi \twoheadrightarrow \psi$. Thus the proof looks as follows:

$$(12) \qquad \ldots, \chi, \ldots, \chi \twoheadrightarrow \psi, \ldots, \psi, \ldots$$

Now by induction hypothesis $\Sigma \vdash \varphi \twoheadrightarrow \chi$ and $\Sigma \vdash \varphi \twoheadrightarrow (\chi \twoheadrightarrow \psi)$. Now,

$$(13) \qquad (\varphi \twoheadrightarrow (\chi \twoheadrightarrow \psi)) \twoheadrightarrow ((\varphi \twoheadrightarrow \chi) \twoheadrightarrow (\varphi \twoheadrightarrow \psi))$$

is a theorem and so we get that $\Sigma \vdash \varphi \twoheadrightarrow \psi$ with two applications of ($\mathrm{MP}_{\twoheadrightarrow}$). ❑

For any given set Σ there exists at most one (finitary and structural) consequence relation $\vdash$ with a deduction theorem for a given connective such that Σ is the set of tautologies of $\vdash$. For assume $\Delta \vdash \varphi$ for a set Δ. Then since $\vdash$ is finitary, there exists a finite set $\Delta_0 \subseteq \Delta$ such that $\Delta_0 \vdash \varphi$. Let $\Delta_0 := \{\delta_i : i < n\}$. Put

$$\text{(14)} \qquad \mathrm{ded}(\Delta_0, \varphi) := \delta_0 \twoheadrightarrow (\delta_1 \twoheadrightarrow \ldots (\delta_{n-1} \twoheadrightarrow \varphi) \ldots)$$

Then, by the deduction theorem for $\twoheadrightarrow$

$$\text{(15)} \qquad \Delta \vdash \varphi \qquad \Leftrightarrow \qquad \varnothing \vdash \mathrm{ded}(\Delta, \varphi)$$

THEOREM 13. *Let $\vdash$ and $\vdash'$ be consequence relations with* $\mathrm{Taut}(\vdash) = \mathrm{Taut}(\vdash')$. *Suppose that there exists a binary term function $\twoheadrightarrow$ such that $\vdash$ and $\vdash'$ satisfy (DT) for $\twoheadrightarrow$. Then $\vdash = \vdash'$.*

2.4 Interpolation

$\vdash$ has **interpolation** if whenever $\varphi \vdash \psi$ there exists a formula χ (called **interpolant**) with $\mathrm{var}(\chi) \subseteq \mathrm{var}(\varphi) \cap \mathrm{var}(\psi)$ such that both $\varphi \vdash \chi$ and $\chi \vdash \psi$. Interpolation is a rather strong property, and generally logics fail to have it. There is a rather simple theorem which allows to prove interpolation for logics based on a finite matrix. Say that $\vdash$ has a **conjunction** if there is a term $p \wedge q$ such that the following are derivable rules: $\langle \{p, q\}, p \wedge q \rangle$ and both $\langle \{p \wedge q\}, p \rangle$ and $\langle \{p \wedge q\}, q \rangle$. In addition, if $\vdash = \vdash_{\mathfrak{M}}$ for some logical matrix $\mathfrak{M} = \langle M, D \rangle$ we say that $\vdash$ has **all constants** if for each $s \in M$ there exists a nullary term function $\underline{s}$ such that for all valuations v $\overline{v}(\underline{s}) = s$. (Note that since $\mathrm{var}(\underline{s}) = \varnothing$ the value of $\underline{s}$ does not depend at all on v.) This rather complicated definition allows that we do not need to have a constant for each truth-value; it is enough if they are definable from the others. For example in classical logic we may have only $\top = \underline{1}$ as a primitive and then $\underline{0} = \neg \top$. An algebra is **functionally complete** if every function $A^n \to A$ is a term function of $\mathfrak{A}$; $\mathfrak{A}$ is **polynomially complete** if every function $A^n \to A$ is a polynomial function. Every functionally complete algebra is polynomially complete; the converse need not hold, since polynomials may employ constants for the elements of $\mathfrak{A}$. However, if $\mathfrak{A}$ has all constants, then it is functionally complete iff it is polynomially complete.

THEOREM 14. *Suppose that $\mathfrak{M}$ is a finite logical matrix. Suppose that $\vdash_{\mathfrak{M}}$ has a conjunction $\wedge$ and all constants; then $\vdash_{\mathfrak{M}}$ has interpolation.*

(See [39], Theorem 1.6.4, where a proof is given.) A property closely related to interpolation is *Halldén-completeness*, named after Sören Halldén, who discussed it first in [31]. (See also [54].) $\vdash$ is called **Halldén-complete** if for all φ and ψ with $\mathrm{var}(\varphi) \cap \mathrm{var}(\psi) = \varnothing$: if $\varphi \vdash \psi$ and φ is consistent then $\vdash \psi$. 2-valued logics are Halldén-complete. Namely, assume that φ is consistent. Let $v : \mathrm{var}(\psi) \to 2$ be a valuation. Since φ is consistent there exists a $u : \mathrm{var}(\varphi) \to 2$ such that $\overline{u}(\varphi) = 1$. Put $w := u \cup v$. Since u and v have disjoint domains, this is well-defined. Then $\overline{w}(\varphi) = 1$, and so $\overline{w}(\psi) = 1$. So, $\overline{v}(\psi) = 1$. This shows that $\vdash \psi$. The following generalisation is now evident.

THEOREM 15 (Łos & Suszko). *Let $\mathfrak{M}$ be a logical matrix. Then $\vdash_{\mathfrak{M}}$ is Halldén-complete.*

In classical logic, the property of Halldén-completeness can be reformulated in a somewhat more familiar form. Namely, the property says that for φ and ψ disjoint in variables, if $\varphi \vee \psi$ is a tautology then either φ or ψ is a tautology.

Finally notice

THEOREM 16. *Suppose that $\mathfrak{M}$ is a logical matrix and $\vdash_{\mathfrak{M}}$ has all constants. Then $\vdash_{\mathfrak{M}}$ is structurally complete and Post-complete.*

2.5 Modal Logics and Modal Consequence Relations

A **modal consequence relation** is a structural consequence relation of modal formulae which contains at least the classical tautologies and in which the rule $(\mathrm{MP}_{\rightarrow})$ is derived. Unless otherwise stated, modal consequence relations are assumed to be finitary. If in addition for every basic modality $\Box$ the rule $(\mathrm{E}_{\Box}) := \langle\{p \leftrightarrow q\}, \Box p \leftrightarrow \Box q\rangle$ is admissible, $\vdash$ is called **classical**. If the rule $(\mathrm{M}_{\Box}) := \langle\{p \rightarrow q\}, \Box p \rightarrow \Box q\rangle$ is admissible for every basic modality $\Box$, $\vdash$ is called **monotone**. Finally, if all rules $(\mathrm{MN}_{\Box}) := \langle\{p\}, \Box p\rangle$ are admissible, $\vdash$ is called **normal**. For simplicity, we refer to the set of the rules $(\mathrm{MN}_{\Box})$, $\Box$ a basic modality as (MN), and treat it (somewhat inappropriately) as a single rule.

Modal logic is typically the study of modal *logics* and not that of modal *consequence relations.* The relationship is one-to-many. If $\vdash$ is a (modal) consequence relation, then

$$(16) \qquad \mathrm{Taut}(\vdash) := \{\varphi : \varnothing \vdash \varphi\}$$

is a modal logic, where a modal logic is any substitution closed set of formulae which contains all classical tautologies and $(\mathrm{MP}_{\rightarrow})$. There is a converse map. Given a logic L, put

$$(17) \qquad \vdash_L := \vdash^{L;(\mathrm{MP}_{\rightarrow})}$$

where L is here identified with the set of rules $\langle\varnothing, \varphi\rangle$, $\varphi \in L$. Evidently, $\Delta \vdash_L \varphi$ iff $\Delta; L \vdash^{(\mathrm{MP}_{\rightarrow})} \varphi$. By Theorem 12 $\vdash_L$ has a DT for $\rightarrow$. We shall often tacitly identify L with $\vdash_L$.

DEFINITION 17. L is **classical** (**monotone**, **normal**) if $\vdash_L$ is. The smallest normal logic with κ operators is denoted by $\mathbf{K}_\kappa$. L is **quasi-normal** if L contains $\mathbf{K}_\kappa$.

We also call a consequence relation **quasi-normal** if its set of tautologies is. Call a term $t(p)$ a **normal operator** for L if it satisfies (a) $t(\varphi \rightarrow \chi) \rightarrow (t(\varphi) \rightarrow t(\chi)) \in L$, and (b) if $t(\varphi) \in L$ then $\Box_i t(\varphi) \in L$. There is a class of formulae that generally are normal if all basic modalities are; these are the so-called compound modalities. A term $t(p)$ with just one variable is called a **compound modality** if it just contains the connectives $\Box_i$, $i < \kappa$ and $\wedge$ in addition to constants; and no variable except for p. One can assign a relation corresponding to $t(p)$ on a frame $\mathfrak{F} = \langle F, \langle \lhd_i : i < \kappa\rangle\rangle$ by induction on its structure as follows.

$$(18) \qquad \begin{aligned} R(p) &:= \{\langle x, x\rangle : x \in F\} \\ R(\Box_i s) &:= \lhd_i \circ R(s) \\ R(s \wedge t) &:= R(s) \cup R(t) \end{aligned}$$

Then for all valuations β and $x \in F$:

$$(19) \qquad \langle \mathfrak{F}, \beta, x\rangle \vDash t(\varphi) \quad \Leftrightarrow \quad \text{for all } y \text{ such that } x\ R(t)\ y\text{: } \langle \mathfrak{F}, \beta, y\rangle \vDash \varphi$$

Let L be a modal logic. Then define

(20) $\mathrm{CRel}(L) := \{\vdash : \mathrm{Taut}(\vdash) = L\}$

Furthermore, let $\vdash_L^m$ be the modal consequence relation containing $\vdash_L$ in which every admissible rule is derived. (It can be obtained by adding to $\vdash_L$ all admissible rules.)

PROPOSITION 18. *Let L be a modal logic. Then*

(21) $\mathrm{CRel}(L) = \{\vdash : \vdash_L \subseteq \vdash \subseteq \vdash_L^m\}$

Moreover, $\vdash_L$ is the unique member of $\mathrm{CRel}(L)$ *having a deduction theorem for $\to$ and $\vdash_L^m$ is the unique member which is structurally complete.*

Now, as is reported in [39], for logics contained in **G.3**, $|\,\mathrm{CRel}(L)| = 2^{\aleph_0}$. However, for tabular logics the situation is actually different (see also Theorem 130 below).

THEOREM 19. *Let L be a tabular modal logic over a finite κ. Then* $\mathrm{CRel}(L)$ *is at most countable, and every member of* $\mathrm{CRel}(L)$*, indeed every extension of $\vdash_L$, is finitely axiomatisable and decidable.*

Proof. First, a tabular logic is finitely axiomatisable. This needs some sophistication. Anticipating the results below, notice first that $\mathcal{V}(L)$ is locally finite. Then, using Corollary 49 we establish that $\mathrm{NExt}(L)$ is continuous, by Theorem 47 that $\mathrm{NExt}(L)$ has a basis, and therefore by Theorem 48 that $\mathrm{NExt}(L)$ has a strong basis. It follows with Theorem 50 that every extension of $\mathrm{NExt}(L)$ is finitely axiomatisable. So, $\vdash_M$ is finitely axiomatisable for every $M \supseteq L$. Also, $\mathcal{V}(L)$ is locally finite. Now, every extension of $\vdash_L$ is determined by some set of matrices verifying the axioms L. This means that they satisfy the axiom that the algebra has at most $|A|$ elements. This reduces the irreducible matrices to those of the form $\langle \mathfrak{B}, D\rangle$ where $|B| \leq |A|$, of which there are only finitely many. (The exact argument is nontrivial, see also [14], Corollary 2.5.20.) Thus, $\vdash_{\mathfrak{A}}$ has finitely many extensions. It is not difficult to show that they are all compact. Being determined by a finite set of finite algebras, they are all decidable. ❑

To see some more examples, consider the rule $\langle\{\Box p\}, p\rangle$. It is admissible in **K**. For assume that $\varphi := p^\sigma$ is not a theorem. Then there exists a model $\langle \mathfrak{F}, \beta, x\rangle \models \neg\varphi$ based on the Kripke-frame $\langle F, \lhd\rangle$. Consider the frame $\mathfrak{G}$ based on $F \cup \{z\}$, where $z \notin F$, and the relation $\blacktriangleleft := \lhd \cup \{\langle z, y\rangle : y \in F\}$. Take $\gamma(p) := \beta(p)$. Then $\langle \mathfrak{G}, \gamma, z\rangle \models \neg\Box\varphi$. The rule $\langle\{p\}, \Diamond p\rangle$ is not admissible in **K** despite the admissibility of $\langle\{\Box p\}, p\rangle$. Take $p := \top$. $\Diamond\top$ is not a theorem of **K**. Similarly, the so-called MacIntosh rule $\langle\{p \to \Box p\}, \Diamond p \to p\rangle$ is not admissible for **K**. Namely, put $p := \Box\bot$. $\Box\bot \to \Box\Box\bot$ is a theorem but $\Diamond\Box\bot \to \Box\bot$ is not. Notice also that if a rule ρ is admissible in a logic L we may not conclude that ρ is admissible in every extension of L. A case in point is the rule $\langle\{\Box p\}, p\rangle$, which is not admissible in $\mathbf{K} \oplus \Box\bot$.

2.6 Lattices of Modal Consequence Relations

Every finitary consequence relation has the form $\vdash^R$ for some set R of finitary rules. Define

(22) $\vdash^R \sqcap \vdash^S := \vdash^R \cap \vdash^S$

(23) $\vdash^R \sqcup \vdash^S := \vdash^{R\cup S}$

We can even define infinitary analogs of the operations:

$$\prod_{i \in I} \vdash_i := \bigcap_{i \in I} \vdash_i \tag{24}$$

$$\bigsqcup_{i \in I} \vdash^{R_i} := \vdash^{(\bigcup_{i \in I} R_i)} \tag{25}$$

It is to be noted, though, that the infinite intersection of finitary consequence relations need not be finitary again. It is also not possible to axiomatize it in terms of the rules for the $\vdash_i$. Therefore in the sequel we shall frequently deal with lattices in which only join is infinitary.

If a finitary rule is derivable in $\vdash^S$, then it is derivable already in $\vdash^{S_0}$ for some finite S_0, since $\vdash^S$ is finitary by assumption. It follows that a consequence relation is compact iff it is finitely axiomatisable. Moreover, the lattice is algebraic, since $\vdash^R = \bigsqcup_{\rho \in R} \vdash^\rho$. Finally, $\vdash'$ is quasi-normal iff $\mathrm{Taut}(\vdash')$ is quasi-normal iff $\mathrm{Taut}(\vdash')$ contains $\mathbf{K}_\kappa$.

PROPOSITION 20. *The set of modal consequence relations over a given language forms an algebraic lattice. The compact elements are exactly the finitely axiomatisable consequence relations. The lattice of quasi-normal consequence relations is the sublattice of consequence relations containing* $\vdash_{\mathbf{K}_\kappa}$.

We write $\mathrm{Ext}(\vdash)$ for the set of extensions of $\vdash$. By abuse of the notation we shall also denote the lattice over this set by $\mathrm{Ext}(\vdash)$. Similarly $\mathrm{QExt}(\vdash)$ denotes the set and the lattice of quasi-normal extensions. $\mathrm{NExt}(L)$ denotes the set and the lattice of normal extensions of a modal logic L.

PROPOSITION 21. *For each quasi-normal logic L and each quasi-normal consequence relation* $\vdash'$,

$$\vdash_L \subseteq \vdash' \quad \Leftrightarrow \quad L \subseteq \mathrm{Taut}(\vdash') \tag{26}$$

$\mathrm{Taut}(-)$ *commutes with infinite intersections,* $\vdash_L$ *with infinite intersections and infinite joins. It follows that* $\mathrm{NExt}(\mathbf{K}_\kappa)$ *is a sublattice of* $\mathrm{Ext}(\vdash_{\mathbf{K}_\kappa})$.

$\mathrm{Taut}(-)$ does not commute with joins. For example, let $\vdash_1 := \vdash^m_{\mathbf{G.3}}$ and $\vdash_2 := \vdash_{\mathbf{K} \oplus \Box\bot}$. Then, by Theorem 28, $\vdash_1$ is maximal, and so $\mathrm{Taut}(\vdash_1 \sqcup \vdash_2) = \mathbf{K} \oplus \bot$. However, $\mathbf{G.3} \sqcup \mathbf{K} \oplus \Box\bot = \mathbf{K} \oplus \Box\bot$.

PROPOSITION 22. *In monomodal logic,* $\vdash_L$ *is maximal iff L is a coatom.*

Proof. Clearly, if $\vdash_L$ is maximal in $\mathrm{Ext}(\vdash_{\mathbf{K}})$, L must be a coatom. To show the converse, we need to show that for a maximal consistent normal logic L, $\vdash_L$ is structurally complete. (It will follow that $\mathrm{CRel}(L)$ has exactly one element.) Now, L is Post-complete iff it contains either the formula $\Box\top$ or the formula $p \leftrightarrow \Box p$. Assume that $\vdash_L$ can be expanded by a rule $\rho = \langle \Delta, \varphi \rangle$. Then, by using the axioms ρ can be transformed into a rule $\rho' = \langle \Delta', \varphi' \rangle$ in which the formulae are nonmodal. (Namely, any formula in a rule may be exchanged by a deductively equivalent formula. Either $\Box\top \in L$ and any subformula $\Box\chi$ may be replaced by $\top$, or $p \leftrightarrow \Box p \in L$ and then $\Box\chi$ may be replaced by χ.) A nonmodal rule not derivable in $\vdash_L$ is also not derivable in its boolean fragment, $\vdash^0_L$. By the maximality of the latter, adding ρ' yields the inconsistent logic. ❑

In polymodal logics matters are a bit more complicated. There exist $2^{\aleph_0}$ logics which are coatoms in $\mathrm{NExt}(\mathbf{K}_2)$ without their consequence relation being maximal. Moreover,

in monomodal logics there exist $2^{\aleph_0}$ maximal consequence relations, which are therefore not of the form $\vdash_L$ (except for the two abovementioned consequence relations). Notice that even though a consequence is maximal iff it is structurally complete and Post-complete, Post-completeness is relative to the derivable rules. Therefore, this does *not* mean that the tautologies form a maximally consistent modal logic.

There is another consequence relation frequently associated with a logic, namely

$$\Vdash_L := \vdash^{L;(\mathrm{MP}_{\rightarrow});(\mathrm{MN})} \tag{27}$$

This is called the **global consequence relation**. Evidently, if (MN) is admissible, the set of tautologies is a normal logic, so $\mathrm{Taut}(\Vdash_L)$ is actually the least normal logic containing L.

PROPOSITION 23. *Let L be a normal logic. Then the following are equivalent.*

1. $\vdash_L = \Vdash_L$.
2. $\Vdash_L$ *admits a deduction theorem for* $\rightarrow$.
3. $L \supseteq \mathbf{K}_\kappa \oplus \{p \rightarrow \Box_j p : j < \kappa\}$.
4. *L is the logic of some set of Kripke-frames containing only one world.*

Clearly, if $\vdash_L \neq \Vdash_L$ then there are several consequence relations for a given logic. We will show now that the converse almost holds. For the purpose of stating the theorem, let $\boxed{\bullet}$ and $\boxed{\circ}$ be the one-point irreflexive and reflexive frame, respectively.

PROPOSITION 24. *Let L be a modal logic. Then the following are equivalent.*

(a) $|\,\mathrm{CRel}(L)| = 1$.

(b) $\vdash_L$ *is structurally complete.*

(c) L is the logic of a single Kripke-frame containing a single world.

(d) L is a fusion of monomodal logics of the frames $\boxed{\bullet}$ *or* $\boxed{\circ}$.

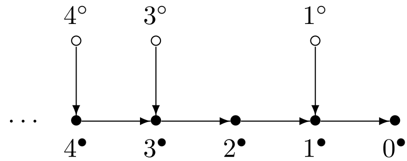
The nontrivial part is to show that (b) $\Leftrightarrow$ (c). Assume (c). Then since $\vdash_L$ is the logic of a single algebra based on two elements, and has all constants, it is structurally complete. Now let (c) fail. There are basically two cases. If L is not the logic of one-point frames, then $\vdash_L$ is anyway not structurally complete by Proposition 22. Otherwise, it is the intersection of logics determined by matrices of the form $\langle \mathfrak{A}, D\rangle$, D an open filter, $\mathfrak{A}$ the free algebra in $\aleph_0$ generators. (In fact, the freely 0-generated algebra is enough.) $\mathfrak{A}$ contains a constant c such that $0 < c < 1$. Namely, take two different one point frames. Then, say, $\Box_0$ is the diagonal on one frame and empty on the other. Then $c := \blacklozenge_0 1$ is a constant of the required form. The rule $\langle\{\Diamond_0\top\}, p\rangle$ is admissible but not derivable.

The method of the last proof can be used in many different ways.

LEMMA 25. *Let L be a logic and χ a constant formula such that neither χ nor $\neg\chi$ are inconsistent. Then the rule $\rho[\chi] := \langle\{\chi\}, \bot\rangle$ is admissible for L but not derivable in $\vdash_L$.*

Since $\chi \notin L$ and $\mathrm{var}(\chi) = \varnothing$, for no substitution s, $\chi^s \in L$. Hence the rule $\rho[\chi]$ is admissible. If it is derivable in $\vdash_L$ then $\vdash_L \chi \rightarrow \bot$, by the DT. So $\neg\chi \in L$, which is not the case. So, $\rho[\chi]$ is not derivable.

Figure 1. $\mathfrak{T}_M$, $M = \{1, 3, 4, \ldots\}$



THEOREM 26. *Let L be a logic such that $\mathfrak{Fr}_L(0)$ has infinitely many elements. Then $|\operatorname{CRel}(L)| = 2^{\aleph_0}$.*

The idea is as follows. There is an infinite set C of constants such that $\chi \wedge \chi' \vdash_L \bot$ whenever χ, χ' are distinct members of C. The relations $\vdash^D$ for $D \subseteq C$ are all pairwise distinct.

COROLLARY 27. *Let L be a monomodal logic and $L \subseteq \mathbf{G.3}$. Then $|\operatorname{CRel}(L)| = 2^{\aleph_0}$.*

In addition, $\vdash^m_{\mathbf{G.3}}$ is maximal. This follows from the following

THEOREM 28. *Let L be the logic of its 0-generated free algebra. Then $\vdash^m_L$ is maximal.*

Proof. Let $\Vdash \supsetneq \vdash^m_L$. Then $\operatorname{Taut}(\Vdash) \supsetneq L$. Since L is determined by its freely 0-generated algebra, there is a constant χ such that $L \subsetneq L \oplus \chi \subseteq \operatorname{Taut}(\Vdash)$. Therefore, we have $\chi \notin L$. (Case 1.) $\neg\chi \notin L$. Then $\rho[\chi]$ is admissible in L and so derivable in $\vdash^m_L$. Therefore $\rho[\chi] \in \Vdash$, and so $\Vdash \bot$. So, $\Vdash$ is inconsistent. (Case 2.) $\neg\chi \in L$. Then $\operatorname{Taut}(\Vdash)$ and also $\Vdash$ is inconsistent. ❑

We will now turn to the set of coatoms in $\operatorname{NExt}(\vdash_{\mathbf{K}})$. Let $M \subseteq \omega$. Put $T_M := \{n^\bullet : n \in \omega\} \cup \{n^\circ : n \in M\}$ and

$$(28) \qquad x \lhd y \quad \Leftrightarrow \quad \left\{ \begin{array}{rl} (1.) & x = m^\bullet, y = n^\bullet \text{ and } m > n \\ \text{or } (2.) & x = m^\circ, y = n^\bullet \text{ and } m \geq n \\ \text{or } (3.) & x = m^\circ, y = n^\circ \text{ and } m = n \end{array} \right.$$

Let $\mathbb{B}_M$ be the algebra of 0-definable sets. Put $\mathfrak{T}_M := \langle T_M, \lhd, \mathbb{B}_M \rangle$. If $M \neq N$ then $\operatorname{Th}(\mathfrak{T}_M) \neq \operatorname{Th}(\mathfrak{T}_N)$. To see this, note that every one-element set $\{n^\circ\}$ in T_M is definable by a formula $\chi(n)$ that depends only on n, not on M. First, take the formula

$$(29) \qquad \delta(n) := \Box^{n+1}\bot \wedge \neg\Box^n\bot$$

$\delta(n)$ defines the set $\{n^\bullet\}$. Now put

$$(30) \qquad \chi(n) := \Diamond\delta(n) \wedge \neg\delta(n+1) \wedge \neg\Diamond\delta(n+1)$$

It is easily checked that $\chi(n)$ defines $\{n^\circ\}$. Hence, if $n \notin M$, $\neg\chi(n) \in \operatorname{Th} \mathfrak{T}_M$. So, $\neg\chi(n) \in \operatorname{Th} \mathfrak{T}_M$ iff $n \notin M$. This establishes that if $M \neq N$, $\operatorname{Th} \mathfrak{T}_M \neq \operatorname{Th} \mathfrak{T}_N$.

THEOREM 29. *The lattice of normal monomodal consequence relations contains $2^{\aleph_0}$ many coatoms.*

2.7 The Locale of Modal Logics—General Theory

Given a normal modal logic L and a set Δ of formulae, $L \oplus \Delta$ denotes the smallest normal logic which contains L and Δ. Recall that $\mathrm{NExt}(L)$ denotes the set (and lattice) of normal logics containing L. For logics $M_i = L \oplus \Delta_i$ we have

(31) $$\bigsqcup_{i \in I} M_i = L \oplus \bigcup_{i \in I} \Delta_i$$

We can also calculate the axiomatisation of the intersection of two logics. Given two formulae, φ and χ, let $\varphi \dot{\vee} \chi$ denote a formula $\varphi \vee \chi^s$, where s is one-to-one and renames the variables of χ so as to make them distinct from the variables of φ. Then

(32) $$(L \oplus \Delta) \sqcap (L \oplus \Sigma) = L \oplus \{\boxplus\varphi \dot{\vee} \boxplus\chi : \varphi \in \Delta, \chi \in \Sigma, \boxplus \text{ a compound modality}\}$$

(See [50] or [39].) This can be used to show that $\mathrm{NExt}(L)$ satisfies the following infinitary distributive law

(33) $$x \sqcap \bigsqcup_{i \in I} y_i = \bigsqcup_{i \in I} x \sqcap y_i$$

In particular, the usual distributivity law holds. This means that the lattice is a locale, where a **locale** is a lattice with infinitary join and finitary meet satisfying (33). Recall that the operation $\bigsqcap$ can be defined from $\bigsqcup$ as follows:

(34) $$\bigsqcap_{i \in I} x_i := \bigsqcup \langle y : \text{for all } i \in I\colon\ y \leq x_i \rangle$$

A locale is **continuous** if also $L \sqcup \bigsqcap_{i \in I} M_i = \bigsqcap_{i \in I} L \sqcup M_i$. Locales $\mathrm{NExt}(L)$ are rarely continuous. An important exception is $\mathrm{NExt}(\mathbf{S4.3})$ (see the remarks following Theorem 47). Call an element x of a locale **meet-irreducible** (**strongly meet-irreducible**) if from $x = y \sqcap z$ follows $x = y$ or $x = z$ (if from $x = \bigsqcap_{i \in I} y_i$ follows $x = y_i$ for some $i \in I$). Call x **meet-prime** (**strongly meet-prime**) if from $x \geq y \sqcap z$ follows $x \geq y$ or $x \geq z$ (if from $x \geq \bigsqcap_{i \in I} y_i$ follows $x \geq y_i$ for some $i \in I$). Dually for **join-irreducible** and **join-prime**. If x is (strongly) meet-prime it is also (strongly) meet-irreducible. In a distributive lattice, meet-prime is equivalent to meet-irreducible, but in general a strongly meet-irreducible element need not be strongly meet-prime. However, in a locale a strongly join-irreducible element is also strongly join-prime.

Given a locale $\mathfrak{L} = \langle L, \sqcap, \bigsqcup \rangle$, let $\mathrm{Irr}(\mathfrak{L})$ be the set of strongly meet-irreducible elements of $\mathfrak{L}$. For $x \in L$ put $x^\circ := \mathrm{Irr}(\mathfrak{L}) - \uparrow x$ where

(35) $$\uparrow x := \{y : y \geq x\} \qquad \downarrow x := \{y : y \leq x\}$$

It turns out that

(36) $$(x \sqcup y)^\circ = x^\circ \cup y^\circ$$

(37) $$(\bigsqcap_{i \in I} x)^\circ = \bigcap_{i \in I} x_i^\circ$$

Thus, $\{x^\circ : x \in L\}$ is a topology of closed sets on $\mathrm{Irr}(\mathfrak{L})$. A locale is **spatial** if it is isomorphic to the locale of open sets of a topological space.

THEOREM 30. *The locale* NExt(L) *is spatial.*

To show that NExt(L) is spatial we need to show that the map $M \mapsto M^\circ$ is injective. For a formula $\varphi \notin M$, the set of logics not containing φ is nonempty (containing, for example, M) and has a maximal element, which we denote by $L^\star_\varphi$. (This follows from Zorn's Lemma, using the fact that NExt(L) is algebraic. $L^\star_\varphi$ is usually not unique.) $L^\star_\varphi$ is easily seen to be strongly meet-irreducible. Now

$$(38) \qquad M = \bigcap_{\varphi \notin M} L^\star_\varphi$$

The topology $\{M^\circ : M \in \mathrm{NExt}(L)\}$ satisfies the T_0-axiom: for every pair M, M' of different logics there is an open set X such that $|X \cap \{M, M'\}| = 1$. Put $M \preccurlyeq M'$ if $M^\circ \subseteq M'^\circ$. It is easy to see that $M \preccurlyeq M'$ iff $M \subseteq M'$ iff $M \leq M'$. Moreover, a closed set is lower closed, that is, if X is closed then $\downarrow X = X$. The converse need not be true. Thus, the lattice is completely reconstructible from the topology. Moreover:

THEOREM 31. NExt(L) *is continuous iff all lower closed sets are closed.*

Indeed, if NExt(L) is continuous, then the arbitrary union of closed sets is closed. Any lower closed set is the union of sets of the form $\downarrow\{x\}$, which are all closed. More on this subject can be found in [39].

It is interesting to know which properties are at all connected with the lattice structure. Completeness, for example, is clearly closed under meet but not under join (for a counterexample see [39]). Elementarity is closed both under intersection and infinitary join. Decidability is closed under intersection, but not under join. Interpolation and Halldén-completeness show no clear connection.

2.8 Splittings

Splittings have been studied in the context of modal logics first by [4], from which most of the results below are drawn. This investigation was carried further in [49, 51]. A **splitting** of a lattice $\langle L, \sqcap, \sqcup\rangle$ is a pair $\langle x, y\rangle$ such that $L = \downarrow x \cup \uparrow y$ and $\downarrow x \cap \uparrow y = \varnothing$. We say that x **splits** $\mathfrak{L}$ if there is y such that $\langle x, y\rangle$ splits $\mathfrak{L}$. We say that y is the **splitting companion** of x and write $\mathfrak{L}/x$ for y (but for logics we write L/M rather than NExt$(L)/M$).

PROPOSITION 32. *If $\langle x, y\rangle$ is a splitting of $\mathfrak{L}$, x is strongly meet-prime and y is strongly join-prime. x splits $\mathfrak{L}$ iff it is strongly meet-prime. If $x < x'$ and $\langle x', y'\rangle$ is a splitting, then $y < y'$.*

Notice that every join-irreducible logic is join-prime. There is a useful corollary for logics. Say that M is **essentially 1-axiomatisable** over L if for every Δ: if $M = L \oplus \Delta$ then there is a $\delta \in \Delta$ such that $M = L \oplus \delta$. It is easy to see that this notion is equivalent to strong join-irreducibility. Hence we have an observation already made in [46].

PROPOSITION 33 (McKenzie). *M is essentially 1-axiomatisable over L iff $M = L/N$ for some splitting logic N.*

Furthermore, this gives rise to an axiomatisability criterion. Suppose that $M = L/N$. Then $M = L \oplus \delta$ iff (a) $\delta \in M$ and (b) $\delta \notin N$. If both M and N are decidable, the problem '$M = L \oplus \delta$' is decidable. For example, $\mathbf{S5} = \mathbf{S4}/N$, where N is the logic of a four element algebra. Then clearly N is decidable; since also $\mathbf{S5}$ is decidable, the problem

'$\mathbf{S5} = \mathbf{S4} \oplus \delta$' is decidable. More can be established. Also the problem '$(\vdash_{\mathbf{S4}})^{+\rho} = \vdash_{\mathbf{S5}}$' is decidable. This is due to the following fact. Recall that $\vdash^m_M$ is the maximal consequence relation that has M as its set of tautologies.

PROPOSITION 34 (Rautenberg). *Suppose that M induces a splitting of* $\mathrm{NExt}(L)$. *Then* $\vdash^m_M$ *splits the lattice* $\mathrm{Ext}(\vdash_L)$, *and* $\mathrm{Ext}(\vdash_L)/\vdash^m_M = \vdash_{L/M}$.

Now, suppose a rule ρ is given. The problem whether $\rho \in \vdash^m_M$ is decidable (see Theorem 19). (Case 1) $\rho \in \vdash^m_M$. Then $(\vdash_{\mathbf{S4}})^{+\rho} \subseteq \vdash^m_M \not\supseteq \vdash_{\mathbf{S5}}$. (Case 2) $\rho \notin \vdash^m_M$. Then $(\vdash_{\mathbf{S4}})^{+\rho} \supsetneq \vdash_{\mathbf{S4}}$. Now we must check whether $\rho \in \vdash_{\mathbf{S5}}$. This is again decidable. If this holds $(\vdash_{\mathbf{S4}})^{+\rho} = \vdash_{\mathbf{S5}}$. The argument generalizes to the case where M is tabular and L/M is decidable.

LEMMA 35. *If M does not split* $\mathrm{NExt}(L)$ *there is a sequence N_i, $i \in \omega$, of logics such that $N_i \not\leq M$ but $\prod_{i\in\omega} N_i \leq M$.*

(And if M does split $\mathrm{NExt}(L)$, no such sequence can obviously exist.) If N splits $\mathrm{NExt}(L)$ it is strongly meet-prime. In particular, it is strongly meet-irreducible. It follows that $N = \mathrm{Th}\,\mathfrak{A}$, where $\mathfrak{A}$ is a subdirectly irreducible (si) algebra. However, there are examples of subdirectly irreducible algebras such that $\mathrm{Th}\,\mathfrak{A}$ is not even meet-irreducible. The algebras that induce splittings can be characterized. Call an element $x < 1$ of an si $\mathfrak{A}$ an **opremum** if for all $a < 1$ there is a compound modality $\boxplus$ such that $\boxplus a \leq x$. Intuitively, in a finite algebra an opremum is easy to find. The dual frame is generated by a single world, w, in the sense that every world is indirectly accessible from it iff the algebra is subdirectly irreducible. (This fails in the infinite, as Giovanni Sambin pointed out, see [39].) Now, the set containing everything but w, is an opremum.

Let $\Delta(\mathfrak{A})$ be the so-called **diagram** of $\mathfrak{A}$, defined by

$$\begin{aligned}(39)\qquad \Delta(\mathfrak{A}) := \quad & \{p_a \vee p_b \leftrightarrow p_{a\vee b} : a, b \in A\}\\ & \cup \{p_{\neg a} \leftrightarrow \neg p_a : a \in A\}\\ & \cup \{p_{\Box_i a} \leftrightarrow \Box_i p_a : a \in A, i < \kappa\}\end{aligned}$$

Suppose that there is an algebra $\mathfrak{B}$, a valuation β, and an ultrafilter U such that $\beta(\neg p_x) \in U$ and for every compound modality $\boxplus$ and every $\delta \in \Delta(\mathfrak{A})$, $\beta(\boxplus\delta) \in U$. Then $\mathfrak{A} \in \mathsf{HS}\,\mathfrak{B}$. Moreover, (by Jónsson's Lemma), $\mathfrak{A} \in \mathsf{HSP}\,\mathfrak{B}$ iff $\mathfrak{A} \in \mathsf{HSUp}\,\mathfrak{B}$ iff every finite subset of $p_x; \bigcup\{\boxplus\Delta(\mathfrak{A}) : \boxplus \text{ a compound modality}\}$ is satisfiable. Let $\mathcal{V}(L)$ denote the variety of L-algebras. The following result appeared in its complete form in [61], generalizing theorems by [51] and [36].

THEOREM 36 (Wolter). *Let $\mathfrak{A}$ be subdirectly irreducible with opremum x. The following are equivalent:*

① $\mathrm{Th}\,\mathfrak{A}$ *splits* $\mathrm{NExt}(L)$.

② *There is a finite $\Delta_0 \subseteq \Delta(\mathfrak{A})$ and a compound modality $\boxplus$ such that for every $\mathfrak{B} \in \mathcal{V}(L)$: if $\neg p_x; \boxplus\Delta_0$ is satisfiable in $\mathfrak{B}$, so is*

$$\neg p_x; \bigcup\{\boxtimes\Delta(\mathfrak{A}) : \boxtimes \textit{ a compound modality}\}$$

If either obtains,

$$(40)\qquad L/\mathrm{Th}\,\mathfrak{A} = L \oplus \bigwedge \boxplus\Delta_0 \to p_x$$

We note that the number of variables needed to axiomatize $L/\operatorname{Th}\mathfrak{A}$ is the minimum number of variables needed to generate $\mathfrak{A}$. This can be used to show that **S4.3** cannot be axiomatized over **S4.2** (and **S4**) using less than two variables. (In tense logic, however, one variable is sufficient.)

2.9 Some Splittings

Let us first look at monomodal logics. A frame $\mathfrak{F}$ is cycle-free if there is an $n \in \omega$ such that $\mathfrak{F} \vDash \Box^n\bot$. **K** is complete with respect to all cycle-free frames. It follows from Lemma 35 that only logics of cycle-free frames can split $\mathrm{NExt}(\mathbf{K})$. So, let $\mathfrak{F}$ be finite, cycle free and generated by a single point. Let $\mathfrak{A}$ be the algebra of its subsets. $\mathfrak{A}$ is si. Since $\mathfrak{A} \vdash \Box^n\bot \to \Box^{n+1}\bot$, it follows that if $\Box^n\bot$ is satisfiable, then $\Box^{n+k}\bot$ is satisfiable for every $k \in \omega$. Thus, since $\Box^n\Delta(\mathfrak{A})$ is finite and implies $\Box^n\bot$, we get that $\Box^n\Delta(\mathfrak{A}) \to \Box^{n+k}\Delta(\mathfrak{A})$ for every k. So, the theory of a finite one-generated cycle free frame splits $\mathrm{NExt}(\mathbf{K})$. This argument generalizes easily for any finite number of operators.

THEOREM 37 (Blok). *L splits* $\mathrm{NExt}(\mathbf{K})$ *iff it is the logic of a finite, one-generated cycle free frame.*

There is an easy corollary. For every splitting logic L there is a splitting logic $L' < L$. (Simply add another irreflexive point before the generator of $\mathfrak{F}$ where $L = \mathrm{Th}(\mathfrak{F})$.) Now, from Proposition 32 we get $\mathrm{NExt}(\mathbf{K})/L' < \mathrm{NExt}(\mathbf{K})/L$. Thus, for every strongly join-prime element there exists a strongly join-prime element strictly below it. Atoms are strongly join-irreducible, and therefore strictly join-prime, hence they are splitting companions. We have established the following result from [3].

THEOREM 38 (Blok). $\mathrm{NExt}(\mathbf{K})$ *is atomless.*

On the other hand we have the following from [41].

THEOREM 39 (Makinson). $\mathrm{NExt}(\mathbf{K})$ *has exactly two coatoms. Moreover, every consistent logic is below one of them.*

The coatoms are the logics of the two two-element algebras, corresponding to the one-element reflexive frame, and the one-element irreflexive frame. Take a general frame $\mathfrak{F}$. Either $\Box\bot$ is satisfiable, in which case the subframe of points satisfying $\Box\bot$ is generated and can be contracted to the single one-generated irreflexive point; or $\Box\bot$ is not satisfiable. Then $\mathfrak{F} \vDash \Diamond\top$, so that $\mathfrak{F}$ is contractible to a one-element reflexive frame. The second fact easily follows from the following observation: if L is finitely axiomatisable, there is no infinite upgoing chain with limit L. The inconsistent logic is finitely axiomatisable, and so it is not the limit of an upgoing chain. Hence every consistent logic must be below a coatom.

Suppose now that $L = \mathbf{K}/M$ for some logic M. It so happens that $\mathrm{NExt}(L)$ may be split by N even though N does not split $\mathrm{NExt}(\mathbf{K})$. This arises exactly once: $L = \mathbf{K}/\boxed{\bullet}$. Then $L = \mathbf{K.D}$, and the new splitting logic is $N = \mathrm{Th}\,\boxed{\circ}$, the logic of the one-element reflexive frame. We call L/N an **iterated splitting** of **K**. L/N is actually inconsistent. However, suppose that X is a set of splitting logics of $\mathrm{NExt}(L)$. Then we may split off the logics of X in any order we like. The results is always the same. Therefore, put

$$L/X := \bigsqcup \langle L/N : N \in X\rangle \tag{41}$$

The following theorem is much harder to establish. Let $\mathrm{Fs}(L)$ be the set of all logics that have the same Kripke-frames as L (the **Fine-spectrum** of L). Call L **intrinsically complete** if $|\,\mathrm{Fs}(L)| = 1$. The following is from [4].

THEOREM 40 (Blok). *L is intrinsically complete iff it is inconsistent or of the form $\mathbf{K}/X$ for a set of splitting logics X. If L is not intrinsically complete, $|\,\mathrm{Fs}(L)| = 2^{\aleph_0}$.*

We say that N has a **splitting representation** over L if it has the form L/X for some set X. Although one can have $N = L/X = L/Y$ for different X and Y, there is a unique set $X^\star$ such that $N = L/X^\star$ and for every X such that $N = L/X$ we have $X \supseteq X^\star$. (The set $X^\star$ is a minimal representation of L.)

Say that a compound modality $\boxplus$ is a **master modality** for L if (a) $\boxplus p \to \Box_i p \in L$ for all $i < \kappa$, and (b) $\boxplus p \to p, \boxplus p \to \boxplus\boxplus p \in L$. L is called **weakly transitive** if it has a master modality. Now suppose that L is weakly transitive, with master modality $\boxplus$. Then if $\mathfrak{A}$ is finite and subdirectly irreducible, it is splitting. (Actually, it is enough that $\mathfrak{A}$ is finitely presentable.) For example, the logic M of a one-generated finite frame splits $\mathrm{NExt}(\mathbf{K4})$ (and every $\mathrm{NExt}(L)$ for $L \geq \mathbf{K4}$ if only $M \geq L$). Many logics above $\mathbf{K4}$ possess a splitting representation above $\mathbf{K4}$.

We present a few applications. [19] shows that there is an infinite antichain L_i, $i \in \omega$, of logics of depth 3 in $\mathrm{NExt}(\mathbf{S4})$. Now, define the following map from subsets of ω into $\mathrm{NExt}(\mathbf{S4})$: $p : U \mapsto \mathbf{S4}/\{L_i : i \in U\}$. This map is injective. Moreover, $p(U) \leq p(V)$ iff $U \subseteq V$. So, the map is a lattice embedding. It follows not only that $\mathrm{NExt}(\mathbf{S4})$ has continuously many elements, but also that it has an infinite upgoing chain of elements.

It is known that every logic $L \supseteq \mathbf{S4.3}$ has the finite model property (see [7]). It follows that it has a representation

$$L = \mathbf{S4.3}/X \tag{42}$$

where X is the set of logics of $\mathbf{S4.3}$-frames which are not L-frames. Identity holds by the fact that both logics have the finite model property and the same finite models. It follows that there is a unique minimal set $X^\star$ such that $L = \mathbf{S4.3}/X^\star$. This means that there is a canonical axiomatisation of every logic in terms of splitting formulae, an axiomatisation base in the sense defined below.

2.10 Axiomatisation Bases

The success of the canonical formulae of Michael Zakharyaschev (see [63, 64]) has sparked off the question whether it is possible to find independent sets of formulae that can axiomatize any given logic above L, where L is a given modal logic (in the best case, $L = \mathbf{K}_\kappa$). The present section reviews conditions on L under which this is possible, but the outcome is, for practical purposes, rather negative: only very strong logics L have this property.

If every extension of L is of the form L/X the locale $\mathrm{NExt}(L)$ is continuous. The finitely axiomatisable logics are closed under finite union, just as the compact elements. An infinite join of finitely axiomatisable logics need not be finitely axiomatisable again. Likewise, the finite meet of finitely axiomatisable logics need not be finitely axiomatisable. However, this is the case when L is weakly transitive.

DEFINITION 41. A locale is **coherent** if (i) every element is the join of compact elements and (ii) the meet of two compact elements is again compact.

Coherent locales allow a stronger representation theorem. Let $\mathfrak{L}$ be a coherent locale, $K(\mathfrak{L})$ be the set of compact elements. They form a lattice $\mathfrak{K}(\mathfrak{L}) := \langle K(\mathfrak{L}), \sqcap, \sqcup \rangle$, by definition of a coherent locale. Given $\mathfrak{K}(\mathfrak{L})$, $\mathfrak{L}$ is uniquely identified by the fact that it is the lattice of ideals of $\mathfrak{K}(\mathfrak{L})$.

LEMMA 42. *A locale is coherent iff it is isomorphic to the locale of ideals of a distributive lattice.*

If we have a lattice homomorphism $\mathfrak{K}(\mathfrak{L}) \to \mathfrak{K}(\mathfrak{M})$ then this map can be extended uniquely to a homomorphism of locales $\mathfrak{L} \to \mathfrak{M}$. Not all locale homomorphisms arise this way, and so not all locale maps derive from lattice homomorphisms. Hence call a map $f : \mathfrak{L} \to \mathfrak{M}$ **coherent** if it maps compact element into compact elements.

THEOREM 43. *The category* DLat *of distributive lattices and lattice homomorphisms is dual to the category* CohLoc *of coherent locales with coherent maps.*

Now if L is weakly transitive, the intersection of two finitely axiomatisable extensions is again finitely axiomatisable. Now, a logic is compact in $\mathrm{NExt}(L)$ iff it is finitely axiomatisable over L. We conclude the following theorem.

PROPOSITION 44. *Let L be weakly transitive. Then* $\mathrm{NExt}(L)$ *is coherent.*

The converse need not hold. $\mathrm{NExt}(\mathbf{K.alt_1}) = \mathbf{K} \oplus \Diamond p \to \Box p$ is coherent (because every logic in this lattice is finitely axiomatisable) but the logic is not weakly transitive.

DEFINITION 45. Let $\mathfrak{L}$ be a complete lattice. A set $X \subseteq L$ is a **generating set** if for every member of L is the join of a subset of X. $\mathfrak{L}$ is said to have a **basis** if there exists a least generating set. Moreover, X is a **strong basis** for $\mathfrak{L}$ if every element has a nonredundant representation, that is, for each x there exists a minimal $Y \subseteq X$ such that $x = \bigsqcup Y$.

THEOREM 46. *Let $\mathfrak{L}$ be a locale. $\mathfrak{L}$ has a basis iff (i) $\mathfrak{L}$ is continuous and (ii) every element is the meet of $\bigsqcap$-irreducible elements. $\mathfrak{L}$ has a strong basis iff it has a basis and there exists no infinite properly ascending chain of $\bigsqcap$-prime elements.*

Let $\mathfrak{L}$ be a locale with a strong basis. Then the elements of $\mathfrak{L}$ are in one-to-one correspondence with antichains of strongly meet-prime elements (via the splitting representation, which must exist).

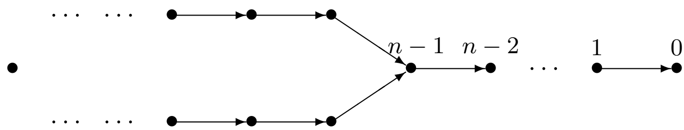
THEOREM 47. *Let L be a modal logic. Then* $\mathrm{NExt}(L)$ *has a basis iff* $\mathrm{NExt}(L)$ *is continuous.*

Since continuous lattices are the exception in modal logic, most extension lattices do not have a basis. We can sharpen the previous theorem somewhat to obtain stricter conditions on continuity. From Theorem 47 and the next theorem it follows that $\mathrm{NExt}(\mathbf{S4.3})$ is continuous (using the result of [7] that all extensions of $\mathbf{S4.3}$ have the fmp).

COROLLARY 48. *Let L be weakly transitive and have the finite model property. Then the following are equivalent.*

1. $\mathrm{NExt}(L)$ *has a basis.*
2. $\mathrm{NExt}(L)$ *has a strong basis.*
3. *Every extension of L has the finite model property.*
4. *Every extension of L is the join of co-splitting logics.*

Figure 2. The Frame $\mathfrak{O}$



5. *Every join of co-splitting logics has the finite model property.*

COROLLARY 49. *Let* $\mathcal{V}(L)$ *be locally finite. Then* $\mathrm{NExt}(L)$ *is continuous.*

The converse does not hold. The lattice $\mathrm{NExt}(\mathbf{S4.3})$ is continuous but **S4.3** fails to be locally finite. The following once more emphasizes the importance of splittings on the structure of the lattice.

THEOREM 50. *Let* $\mathrm{NExt}(L)$ *have a strong basis. Then the following are equivalent.*

1. *Every extension of* $\mathrm{NExt}(L)$ *is finitely axiomatisable.*
2. $\mathrm{NExt}(L)$ *is finite or countably infinite.*
3. *There exists no infinite set of incomparable splitting logics.*

Typically the locales $\mathrm{NExt}(L)$ have no basis. We might ask, however, if for a given logic an independent axiomatisation necessarily exists. This is not so. Call a set Δ of formulae **independent** if for every $\delta \in \Delta$ we have $\delta \notin \mathbf{K} \oplus (\Delta - \{\delta\})$. (For example, a basis is an independent set.) A logic L is **independently axiomatisable** if there exists an independent set Δ such that $L = \mathbf{K} \oplus \Delta$. Every finitely axiomatisable logic is independently axiomatisable. It has been shown in [8] that there exists a logic which is not independently axiomatisable. Furthermore, [37] gives an example of a logic which is not finitely axiomatisable, but all its proper extensions are. Such a logic is called **pre-finitely axiomatisable**. Here is a logic that has both properties.

THEOREM 51. *The logic of the frame* $\mathfrak{O}$ *shown in Figure 2 is pre-finitely axiomatisable. It splits the lattice of extensions of* $\mathbf{G}.\Omega_2$*. Moreover, it is not axiomatisable by a set of independent formulae.*

THEOREM 52. *Let* $\mathfrak{A}$ *be the algebra generated by the singleton sets of* $\mathfrak{O}$*.* $\mathfrak{A}$ *is not finitely presentable. Its logic splits* $\mathrm{NExt}(\mathbf{G}.\Omega_2)$*.*

3 THE LOCAL AND THE GLOBAL

3.1 Equivalential and Algebraisable Logics

In recent years, there have been a lot of results concerning the algebraisability of logics. (See [14] for a general exposition of the topics of this section.) Research has been sparked off mainly by the monograph [5]. In brief, a logic is algebraisable if the notion of truth

and of consequence can be reduced faithfully to the equational calculus. Let us assume that two consequence relations, $\vdash$ over language $\mathcal{L}_1$ and $\succ$ over language $\mathcal{L}_2$, are given. Let $\kappa : \mathcal{L}_1 \to \wp(\mathcal{L}_2)$ be a map from formulae in $\mathcal{L}_1$ to sets of formulae in $\mathcal{L}_2$. We write $\kappa(\Delta)$ for the union of the $\kappa(\delta)$, $\delta \in \Delta$. κ is a **transform** of $\vdash$ into $\succ$ if

(43) $$\Delta \vdash \varphi \quad \Leftrightarrow \quad \text{for all } \chi \in \kappa(\varphi) : \kappa(\Delta) \succ \chi$$

If $\lambda : \mathcal{L}_2 \to \wp(\mathcal{L}_1)$ is a transform of $\succ$ into $\vdash$ and κ a transform of $\vdash$ into $\succ$ we call $\langle \kappa, \lambda \rangle$ a **pair of conjugate transforms** if in addition

(44) $$\varphi \vdash \chi \Leftrightarrow \lambda(\kappa(\varphi)) \vdash \chi$$

(45) $$\varphi \succ \chi \Leftrightarrow \kappa(\lambda(\varphi)) \succ \chi$$

A consequence relation is **algebraisable** if there is a pair of conjugate transforms to a calculus of equations over the same language, and both maps commute with substitutions. Recall that there is also a first-order theory of the algebra, using the function symbols of the signature and equality (=). In equational logic we are mainly interested in Horn-clauses of that languages, to which we turn below.

A key element in the characterisation of algebraisability is that of the Leibniz operator. A logic $\vdash$ defines the following operator $\Omega_{\mathfrak{A}}$ on an algebra $\mathfrak{A}$, called the **Leibniz operator**.

(46) $$\Omega_{\mathfrak{A}}(D) := \{\langle a, b \rangle : \text{for all polynomials } p \text{ of } \mathfrak{A} : p(a) \in D \Leftrightarrow p(b) \in D\}$$

Given D, $\Omega_{\mathfrak{A}}$ is the largest congruence compatible with D. $\langle \mathfrak{A}/\Omega_{\mathfrak{A}}(D), D/\Omega_{\mathfrak{A}} \rangle$ is reduced. We write Ω for the operator defined on the term algebra. As Wim Blok and Don Pigozzi have shown, many properties of the consequence relation can be defined in terms of the Leibniz-operator.

THEOREM 53 (Blok & Pigozzi). *A consequence relation $\vdash$ is algebraisable iff*

① Ω *is monotone on the set of theories of* $\vdash$*;*

② Ω *is injective on the set of theories of* $\vdash$*; and*

③ Ω *commutes with inverse substitutions on the set of theories of* $\vdash$*.*

The first is to be read as follows: if T and T' are theories (deductively closed sets of formulae) and $T \subseteq T'$ then $\Omega(T) \subseteq \Omega(T')$. The latter are congruences. Similarly for the other conditions.

We shall fill the notion of algebraisability with more life. The calculus of equations can be generalized to implications. A **quasi-equation** or **quasi-identity** is an implication of the form

(47) $$\sigma_0 = \tau_0 \wedge \sigma_1 = \tau_1 \wedge \ldots \wedge \sigma_{n-1} = \tau_{n-1} \to \sigma_n = \tau_n$$

Alternatively, it is a Horn-clause in the first-order theory of the algebraic signature. A class of algebras is called a **quasi-variety** if it is characterized by a set of quasi-identities. The following is from [30].

THEOREM 54 (Graetzer & Lakser). *A class of algebras is a quasi-variety iff it is closed under ultraproducts, products and subalgebras. The least quasi-variety containing a given class* $\mathcal{K}$ *is* $\mathsf{SPP_u}(\mathcal{K})$.

Now, in general consequence relations use the notion of truth. They are therefore said to **define truth implicitly** if for every algebra $\mathfrak{A}$ there is at most one deductively closed set D such that $\langle \mathfrak{A}, D\rangle$ is a reduced matrix for $\vdash$. An explicit definition consists in a set $\Delta(p)$ of equations such that $a \in D$ iff $\mathfrak{A} \vDash \alpha(a) = \beta(a)$ for all $\alpha(p) = \beta(p) \in \Delta(p)$. Since $\langle \mathfrak{A}, \{1\}\rangle$ is a matrix for all modal consequence relations, and reduced, a consequence relation defines truth implicitly iff (MN) is derivable.

The following definition is due to [48].

DEFINITION 55. Let $\vdash$ be a consequence relation. A set of formulae $\Delta(p,q) := \{\delta_i(p,q) : i \in I\}$ is called a **set of equivalential terms for** $\vdash$ if the following holds for all basic function symbols f:

$$\text{(48a)} \qquad \vdash \Delta(p,p)$$

$$\text{(48b)} \qquad \Delta(p,q) \vdash \Delta(q,p)$$

$$\text{(48c)} \qquad \Delta(p,q); \Delta(q,r) \vdash \Delta(p,r)$$

$$\text{(48d)} \qquad \bigcup_{i<\nu(f)} \Delta(p_i,q_i) \vdash \Delta(f(\vec{p}), f(\vec{q}))$$

$$\text{(48e)} \qquad p; \Delta(p,q) \vdash q$$

$\vdash$ is called **equivalential** if it has a set of equivalential terms, and **finitely equivalential** if it has a finite set of equivalential terms.

THEOREM 56. $\vdash$ *is finitary and finitely equivalential iff the class of reduced matrices for* $\vdash$ *is a quasi-variety.*

COROLLARY 57. *Let* $\vdash$ *be finitary and finitely equivalential and* $\mathcal{Q}$ *the quasi-variety of reduced matrices for* $\vdash$*. Then the lattice of finitary extensions of* $\vdash$ *is dually isomorphic to the lattice of sub-quasi-varieties of* $\mathcal{Q}$*.*

A logic is algebraisable in the sense of Blok and Pigozzi if it is finitary, algebraisable and finitely equivalential. Ω is said to be **continuous** if for every upgoing chain T_i, $i \in \mu$, of theories whose limit (= union) is a theory

$$\text{(49)} \qquad \Omega(\bigcup_{i\in\mu} T_i) = \bigcup\{\Omega\, T_i : i \in \mu\}$$

Continuity implies monotonicity.

THEOREM 58. $\vdash$ *is equivalential iff* Ω *is monotone on the set of theories, and* $s\Omega(T) \subseteq \Omega((sT)^{\vdash})$ *for all substitutions* s *and theories* T*.* $\vdash$ *is finitely equivalential iff* Ω *is continuous on the set of theories of* $\vdash$*.*

Clearly, for any modal logic L, $\vdash_L$ is always equivalential; a set of equivalential terms is the following.

$$\text{(50)} \qquad \Delta(p,q) := \{\boxplus(p \leftrightarrow q) : \boxplus \text{ a compound modality}\}$$

$\Vdash_L$ is always finitely equivalential; $p \leftrightarrow q$ is an equivalential term for $\Vdash_L$. Note that if a classical consequence relation $\vdash$ is finitely equivalential it also has an equivalential term. For if $\Delta(p,q) = \{\delta_i(p,q) : i < n\}$ is a finite set of equivalential terms for $\vdash$ then $\delta(p,q) := \bigwedge_{i<n} \delta_i(p,q)$ is an equivalential term.

For algebraisability in the Blok and Pigozzi sense we have the following.

THEOREM 59 (Blok & Pigozzi). *Let $\vdash$ be algebraisable in the sense of Blok and Pigozzi and let $\mathcal{K}$ be the corresponding class of algebras. Then $\mathcal{K}$ is a quasi-variety and consists of all reducts of reduced matrices. Moreover, the lattice of axiomatic strengthenings is dually isomorphic to the lattice of sub-quasi-varieties of $\mathcal{K}$.*

It is to be borne in mind that there is a substantial difference between classes of matrices and classes of algebras.

3.2 Global Consequence Relations and Logics

Call a modal consequence relation **global** if the rules (MN) are derived rules. If $\vdash$ is global, then any extension contains (MN), and is also global, by structurality. Hence the lattice of global consequence relations is the lattice of extensions of $\Vdash_{\mathbf{K}}$. A modal consequence relation $\vdash$ is finitely equivalential via $p \leftrightarrow q$ iff it is global; in general, other equivalential formulae might exist, see below. A filter D for a consequence relation $\vdash$ in a modal algebra is a boolean filter. However, if $\vdash$ is global, then every filter D is **open**, that is, if $a \in D$ also $\Box a \in D$ for every modality $\Box$. If D is open, it can be factored, and the factor algebra is unital. Hence, reduced matrices for global consequence relations have only one truth value, namely 1. It follows that truth is defined implicitly—and also explicitly via the equation $p = \top$. Thus, we can replace talk of reduced matrices with talk of algebras.

THEOREM 60. *The lattice of global consequence relations is dually isomorphic to the lattice of quasi-varieties of modal algebras.*

Josep Font and Ramon Jansana [21] have found a way to characterize the strong consequence using the Leibniz operator. Say that a filter F on $\mathfrak{A}$ for $\vdash$ is **Leibniz** if for every $\vdash$-filter $G \subseteq F$, $\Omega_{\mathfrak{A}}(G) = \Omega_{\mathfrak{A}}(F)$. The strong consequence relation corresponding to $\vdash$ is the consequence determined by all matrices $\langle \mathfrak{A}, F\rangle$, where $\langle \mathfrak{A}, F\rangle$ is a matrix for $\vdash$ and F is a Leibniz filter. Given any filter, the largest Leibniz filter contained in F is the intersection of all filters G such that $\Omega_{\mathfrak{A}}(G) = \Omega_{\mathfrak{A}}(F)$. In the present context, this filter is the largest open filter contained in F. It consists of all elements a such that $\boxplus a \in F$ for every compound modality.

There is a difference, though, between quasi-varieties of matrices (to be considered below) and quasi-varieties of algebras. The local and global consequence relations for a logic can be characterized as follows.

THEOREM 61.

① $\Delta \vdash_L \chi$ *iff for every generalized frame $\mathfrak{F}$ such that $\mathfrak{F} \vDash L$, every valuation β and every x: if $\langle \mathfrak{F}, \beta, x\rangle \vDash \delta$ for every $\delta \in \Delta$ then $\langle \mathfrak{F}, \beta, x\rangle \vDash \chi$.*

② $\Delta \Vdash_L \chi$ *iff for every generalized frame $\mathfrak{F}$ such that $\mathfrak{F} \vDash L$, and every valuation β: if $\langle \mathfrak{F}, \beta\rangle \vDash \delta$ for every $\delta \in \Delta$ then $\langle \mathfrak{F}, \beta\rangle \vDash \chi$*

Alternatively, $\Delta \Vdash_L \chi$ if for every algebra $\mathfrak{A} \in \mathcal{V}(L)$ and every valuation β: if $\beta(\delta) = 1$ for every $\delta \in \Delta$ then $\beta(\chi) = 1$.

$\vdash_L$ has a deduction theorem but generally, $\Vdash_L$ does not. If it does, however, the logic is weakly transitive.

PROPOSITION 62. *Suppose that $\boxplus$ is a master modality for L. Then $\langle \mathfrak{F}, \beta, x\rangle \vDash \boxplus\chi$ iff χ is true in the model generated by x.*

THEOREM 63. *$\Vdash_L$ has a deduction theorem iff L is weakly transitive.*

The notion of weak transitivity originated in the work of Wim Blok. In weakly transitive logics, the global consequence can be reduced to the local consequence. For $\vdash_L$ is finitely equivalential if L is weakly transitive. Let $\mathrm{Cg}^{\mathfrak{A}}(a, b)$ denote the least congruence of $\mathfrak{A}$ containing the pair $\langle a, b\rangle$. Say that $\mathcal{V}$ has **elementarily definable principal congruences** if there is a first order formula $\vartheta(x, y, u, v)$ such that for all $\mathfrak{A} \in \mathcal{V}$ and $a, b, c, d \in A$, $c\ \mathrm{Cg}^{\mathfrak{A}}(a, b)\ d$ iff $\mathfrak{A} \vDash \vartheta(a, b, c, d)$. Say that $\mathcal{V}$ has **elementarily definable open filters** if there is a first order formula $\eta(x, u)$ such that for given a, c is in the open filter generated by a iff $\mathfrak{A} \vDash \eta(a, c)$. In [6] we find the following.

THEOREM 64. *The following are equivalent.*

① *$\Vdash_L$ has a deduction theorem.*

② *L is weakly transitive.*

③ *L is finitely equivalential.*

④ *$\mathcal{V}(L)$ has elementarily definable principal congruences.*

⑤ *$\mathcal{V}(L)$ has elementarily definable open filters.*

3.3 Semisimple Varieties of Modal Algebras

Semisimple varieties of modal algebras are special kinds of varieties of weakly transitive logics. There is an exact characterisation of semisimplicity, to be found below. Say that $\blacklozenge$ (the diamond of some compound modality $\blacksquare$) is a **dual** of $\Box$ in L if $p \to \Box\blacklozenge p \in L$. Frame theoretically this means that if $x\ R(\Box)\ y$ then $y\ R(\blacksquare)\ x$. If $\blacksquare$ is compound, $R(\blacksquare)$ is a finite set of finite paths in the frame. A logic is **cyclic** if every basic modality $\Box_i$ has a dual. Notice that the dual need not be basic (although a basic modality playing the role of the dual can be added conservatively). If L is cyclic, also every compound modality has a dual.

LEMMA 65. *If L is cyclic then every finite subdirectly irreducible algebra of $\mathcal{V}(L)$ is simple.*

There are infinite algebras that are si but not simple. For example, take the set of integers and put $x \lhd y$ iff $|x - y| = 1$. Finally, let $\mathbb{O}$ be the set of finite and cofinite elements. The logic of $\mathfrak{Z} := \langle \mathbb{Z}, \lhd, \mathbb{O}\rangle$ is cyclic (with $\Diamond$ the dual of $\Box$), $\mathfrak{Z}$ is si (with opremum $\mathbb{Z} - \{0\}$), but not simple. For the set of cofinite subsets is an open filter.

Call a variety **semisimple** if every si algebra is simple. Further, say that a ternary term $t(x, y, z)$ is a **ternary discriminator** for $\mathfrak{A}$ if for all $a, b, c \in A$: $t(a, b, c) = c$ if $a = b$, and $t(a, b, c) = a$ if $a \neq b$. (See also Chapter 6 on this notion.) A variety $\mathcal{V}$ is a discriminator variety if there is a ternary term $t(x, y, z)$ which is a discriminator for all subdirectly irreducible members of $\mathcal{V}$. Notice that if t is a ternary discriminator, then $u(x) := \neg t(1, x, 0)$ has the property that $u(x) = 1$ if $x = 1$ and $u(x) = 0$ otherwise. (This is the dual notion of the one commonly used.) $u(x)$ is called a **unary discriminator**. If L is weakly transitive it has a master modality $\boxplus$. If it is also cyclic $\boxplus$ has a dual

$\neg \boxtimes \neg$. We can actually assume that $\boxtimes = \boxplus$. Now look at $u(x) := \boxplus x$. By weak transitivity, the open filter generated by $a \in A$ is $\uparrow \boxplus a$. Assume that $a = \boxplus a$. Then $\boxplus \neg \boxplus a = \boxplus \neg \boxplus \neg\neg a \geq \neg a = \neg \boxplus a$, by our assumptions. So, $\uparrow \neg \boxplus a$ also is an open filter. Say that a is **open** if $a = \Box_i a$ for all basic $\Box_i$. The open elements form a boolean algebra. It follows that every si algebra is simple. It also follows that $u(x) := \boxplus x$ is a unary discriminator. The converse is much harder to establish, see [35].

THEOREM 66 (Kracht & Kowalski). *The following are equivalent for modal logics with finitely many operators.*

1. *$\mathcal{V}(L)$ is semisimple.*
2. *$\mathcal{V}(L)$ is a discriminator variety.*
3. *L is weakly transitive and cyclic.*

The remaining part is (1) $\Rightarrow$ (3). Moreover, if a semisimple variety is weakly transitive, cyclicity is easy to show (because both mean that one-generated is the same as connected). So the hard part is to show that semisimple varieties are weakly transitive. We assume that the basic operators are $\Box_i$, $i < n$, and put

$$\Box a := a \wedge \bigwedge_{i<n} \Box_i a \tag{51}$$

The proof is rather involved. It proceeds by first showing that all semisimple varieties of finite type of modal algebras satisfy the property (52) for $r = k$ and $l = 0$.

(52) For every $k \in \omega$ there are $r, l \in \omega$ such that $\mathcal{V} \vDash x \leq \Diamond^l \Box^k \Diamond^r x$.

Note that this is weaker than cyclicity. Now we assume that $\mathcal{V}$ satisfies (52). Define $r(i)$ to be the smallest number such that there exists an $l \in \omega$ with $\mathcal{V} \vDash \Diamond^l \Box^i \Diamond^{r(i)} x \leq x$. The function r is increasing. We define $l(i)$ to be the smallest number such that $\mathcal{V} \vDash \Diamond^{l(i)} \Box^i \Diamond^{r(i)} x \leq x$. Thus, l depends on i via $r(i)$. If $\mathcal{V}$ falsifies $\Diamond^{n+1} x = \Diamond^n x$ for each $n \in \omega$, then for each $i \in \omega$ there is a simple algebra $\mathfrak{A}_i$ in $\mathcal{V}$ and $a_i \in A_i$ such that $\Diamond^{r(i)} a_i < 1$ but $\Diamond^{r(i)+1} a_i = 1$. Now put $b_i := \neg \Diamond^{r(i)} a_i$ and fix an arbitrary $k \in \omega$. Then the following lemma holds.

LEMMA 67. *For every $i \geq k$, we have: $\Diamond^k b_i < 1$ and $\Diamond^{l(k)+r(k)+1} \neg \Diamond^k b_i = 1$.*

Using ultraproducts one obtains an algebra $\mathfrak{B}$ and an element b such that

LEMMA 68. *In $\mathfrak{B}$, for any $k \in \omega$ we have: $\Diamond^k b < 1$ and $\Diamond^{l(k)+r(k)+1} \neg \Diamond^k b = 1$.*

Let $\mathfrak{A} \in \mathcal{V}$ be such that there is a nonzero $a \in A$ with $\Diamond^n a < 1$ for every $n \in \omega$. For instance the free algebra $\mathfrak{Fr}_L(1)$ is such an algebra, as otherwise $\mathcal{V}$ would satisfy $\Diamond^n x = 1$ for some $n \in \omega$. Let $\alpha := \mathrm{Cg}^{\mathfrak{A}}(a, 0)$. α is neither full nor the diagonal. As α is principal, α must have a lower neighbour β in $\mathrm{Cg}(\mathfrak{A})$.

LEMMA 69. *For every congruence β with $\beta \prec \alpha$, there is an $m \in \omega$ such that:*

1. *$\Diamond^{m+1} a \equiv_\beta \Diamond^m a$, and*
2. *$\neg \Diamond^m a \equiv_\beta \Diamond \neg \Diamond^m a$.*

Proof. Let $\Gamma := \{\theta \in \mathrm{Cg}(\mathfrak{A}) : \theta \geq \beta, \theta \not\geq \alpha\}$. If $\Gamma = \{\beta\}$, then $\mathfrak{A}/\beta$ is si but not simple, which cannot be. Thus there is a $\theta \in \Gamma - \{\beta\}$. By congruence distributivity, $\gamma := \bigvee \Gamma \in \Gamma$. Therefore, $\mathfrak{A}/\gamma$ is subdirectly irreducible; hence simple. From this and congruence permutability it follows that $\alpha \circ \gamma = A \times A$. Thus, $(0,1) \in \alpha \circ \gamma$, and there must be a $c \in A$ with $(0,c) \in \alpha$ and $(c,1) \in \gamma$; hence also $(\neg c, 0) \in \gamma$. Now, $(0,c) \in \alpha$ iff for some $m \in \omega$ we have $\Diamond^m a \geq c$. Thus, $\neg\Diamond^m a \leq \neg c$ and therefore $(\neg\Diamond^m a, 0) \in \gamma$. We can then assume $c = \Diamond^m a$. By definition we have $\alpha \cap \gamma = \beta$, that is, $0/\alpha \cap 0/\gamma = 0/\beta$. Now, to prove (i), consider $\Diamond^{m+1} a \wedge \neg\Diamond^m a$. It belongs to $0/\alpha \cap 0/\gamma = 0/\beta$ and thus we obtain $\Diamond^{m+1} a \equiv_\beta \Diamond^m a$. Then, for (ii), consider $\Diamond^m a \wedge \Diamond\neg\Diamond^m a$. It too belongs to $0/\alpha \cap 0/\gamma = 0/\beta$; therefore $\neg\Diamond^m a \equiv_\beta \Diamond\neg\Diamond^m a$. ❑

THEOREM 70. *If $\mathcal{V}$ satisfies* (52) *then $\mathcal{V}$ satisfies $\Diamond^{n+1} x = \Diamond^n x$ for some $n \in \omega$.*

Proof. Suppose $\mathcal{V}$ falsifies $\Diamond^{n+1} x = \Diamond^n x$ for all $n \in \omega$. There is then an algebra $\mathfrak{B} \in \mathcal{V}$ and an element $b \in B$ such that for all $k \in \omega$: $\Diamond^k b < 1$ and $\Diamond^{l(k)+r(k)+1}\neg\Diamond^k b = 1$. Let α be the congruence generated by $\neg b$, and take β and m as in Lemma 69. Then $\neg\Diamond^m b \equiv_\beta \Diamond\neg\Diamond^m b \equiv_\beta \Diamond^{l(m)+r(m)+1}\neg\Diamond^m b = 1$. Thus, $\Diamond^m b \equiv_\beta 0$ and therefore $b \equiv_\beta 0$. It follows that $\beta \geq \alpha$, contradicting the choice of β as a subcover of α. ❑

4 REDUCTION TO MONOMODAL LOGIC

For each cardinality κ, there is a distinct lattice of modal consequence relations over κ operators. Surely, it would be most advantageous if one did not have to study these lattices for each individual κ. While results for the lattices $\mathrm{Ext}(\vdash_{\mathbf{K}_\kappa})$ are yet to be established, there exists fairly powerful theorems that reduce the study of $\mathrm{NExt}(\mathbf{K}_\kappa)$ for finite κ to the study of $\mathrm{NExt}(\mathbf{K}_1)$. It turns out that the locales of logics for several operators are isomorphic to certain subintervals of the locale $\mathrm{NExt}(\mathbf{K}_1)$ and that the isomorphism reflects and preserves many important properties of logic. This means that from a general perspective it is enough to obtain results for the locale of monomodal logics. A theorem that asserts this is called a *transfer theorem*. Results on monomodal logics can be extended to polymodal logics, using a transfer theorem. In practice it has turned out to be the opposite, however. Often, a counterexample to a specific conjecture can be easily constructed using several operators. Using the transfer theorem this counterexample typically yields a counterexample for monomodal logic, and so for every polymodal logic. There are certain lacunae in the theory. First, although there is a simulation of countably many operators by one (see [38]), the induced lattice map is not surjective. As for uncountably many operators, I know of no results so far. The techniques have been applied to polyadic operators and hybrid logics, and we report the results below. Again, the lattice map is not surjective, making the transfer theory less effective. Third, as we have mentioned above, the results cover logics only; no attempt has been made to reduce polymodal consequence relations to monomodal ones, though I speculate that the results will be similar.

4.1 Simulating Two Operators by One

Let $\langle F, \lhd, \blacktriangleleft, \mathbb{F}\rangle$ be a generalized bimodal frame. For a subset $B \subseteq F$ put $B_\circ := \{x_\circ : x \in B\}$ and $B_\bullet := \{x_\bullet : x \in B\}$. Here, $x_\circ$ and $x_\bullet$ are distinct copies of x, $F_\circ$ and $F_\bullet$ are

disjoint and do not contain $*$.

$$
\begin{array}{rl}
F^s := & F_\circ \cup F_\bullet \cup \{*\} \\
\leqslant := & \{\langle x_\circ, y_\circ\rangle : x \lhd y\} \cup \{\langle x_\bullet, y_\bullet\rangle : x \blacktriangleleft y\} \\
& \cup\{\langle x_\circ, x_\bullet\rangle : x \in F\} \cup \{\langle x_\bullet, x_\circ\rangle : x \in F\} \\
& \cup\{\langle x_\circ, *\rangle : x \in F\} \\
\mathbb{F}^s := & \{B_\circ \cup C_\bullet \cup D : B, C \in \mathbb{F}, D \subseteq \{*\}\} \\
\mathfrak{F} := & \langle F^s, \leqslant, \mathbb{F}^s\rangle
\end{array} \tag{53}
$$

$\mathfrak{F}^s$ is a general monomodal frame. We call it the **simulating frame** of $\mathfrak{F}$. Recall that a general frame $\mathfrak{F}$ is **differentiated** if $x \neq y$ implies $x \in a$ and $y \notin a$ for some $a \in \mathbb{F}$; that $\mathfrak{F}$ is **refined** if it is differentiated and **tight**, that is, if $x \ntriangleleft_i y$ then there is an $a \in \mathbb{F}$ such that $x \in \Box_i a$ but $y \notin a$. Finally, $\mathfrak{F}$ is **compact** if for every filter H on $\mathbb{F}$: $\bigcap H \neq \varnothing$.

PROPOSITION 71. *$\mathfrak{F}^s$ is differentiated (refined, compact) iff $\mathfrak{F}$ is.*

Proof. Notice that $F_\circ$, $F_\bullet$ and $\{*\}$ are definable by the constant formulae $\gamma_\circ := \Diamond \boxminus \bot$, $\gamma_\bullet := \Diamond\top \wedge \neg\Diamond \boxminus \bot$, and $\gamma_* := \boxminus\bot$, respectively. (We shall also denote the sets defined by some formula by the formula itself.) Hence if $\mathfrak{F}$ is differentiated, let $x, y \in F^s$ be different. Then if $x = *$, γ_* is the set that contains x but not y. Otherwise if $x = u_\circ$ and $y = v_\bullet$, $\gamma_\circ$ contains x but not y. Finally, if $x = u_\circ$ and $y = v_\circ$ then $u \neq v$ and there is a set O containing x but not y. Then $x \in O_\circ$, but $y \notin O_\circ$. Analogously if $x = u_\bullet$ and $y = v_\bullet$. Also, if $\mathfrak{F}^s$ is differentiated, clearly $\mathfrak{F}$ is differentiated, too. We show that if $\mathfrak{F}$ is refined, so is $\mathfrak{F}^s$. Suppose that $x \leqslant y$ does not hold. The case $x = *$ is easily dealt with. Now assume $x = u_\circ$. We deal with two representative cases. Case 1. $y = v_\circ$. Then $u \ntriangleleft v$. Then refinedness of $\mathfrak{F}$ gives a set O such that $u \in \Box O$ but $v \notin O$. Then $x \in \boxminus(F_\bullet \cup \{*\} \cup O_\circ)$ but $y \notin O_\circ$. Case 2. $y = v_\bullet$. Then $u \neq v$. Then by differentiatedness there is a set O containing u but not v. Then $x = u_\circ \in \boxminus(F_\circ \cup \{*\} \cup O_\bullet)$ but $y \notin O_\bullet$. (Notice that for the transfer of tightness we needed differentiatedness as well.) Transfer of compactness is straightforward. ❑

The notion of simulation is then also defined for Kripke-frames. Denote by $\mathfrak{F}_\sharp$ the Kripke-frame underlying $\mathfrak{F}$. The following is true in virtue of the definitions.

PROPOSITION 72. $(\mathfrak{F}^s)_\sharp = (\mathfrak{F}_\sharp)^s$.

Define

$$
\boxminus_\circ \chi := \boxminus(\gamma_\circ \to \chi) \qquad \boxminus_\bullet \chi := \boxminus(\gamma_\bullet \to \chi) \qquad \boxminus_* \chi := \boxminus(\gamma_* \to \chi) \tag{54}
$$

Also, put

$$
\begin{array}{rl}
p_i^s := & p_i \\
(\neg\varphi)^s := & \neg\varphi)^s \\
(\varphi \wedge \chi)^s := & \varphi^s \wedge \chi^s \\
(\Box\varphi)^s := & \boxminus_\circ \varphi^s \\
(\blacksquare\varphi)^s := & \boxminus_\bullet \boxminus_\bullet \boxminus_\circ \varphi^s
\end{array} \tag{55}
$$

Finally, let β be a valuation and set $\beta^s(p) := \beta(p)_\circ$. Then the following is shown by induction.

$$
\langle \mathfrak{F}, \beta, x\rangle \vDash \varphi \Leftrightarrow \langle \mathfrak{F}^s, \beta^s, x_\circ\rangle \vDash \gamma_\circ \to \varphi^s \tag{56}
$$

Notice that $\langle \mathfrak{F}^s, \beta^s, x_\bullet \rangle \vDash \gamma_\circ \to \varphi^s$ as well as $\langle \mathfrak{F}^s, \beta^s, * \rangle \vDash \gamma_\circ \to \varphi^s$. Not every valuation into $\mathfrak{F}^s$ is of the form β^s. However, if $\gamma(p) \cap F_\circ = \delta(p) \cap F_\circ$ then for every $x \in F^s$

(57) $$\langle \mathfrak{F}^s, \gamma, x \rangle \vDash \gamma_\circ \to \varphi^s \Leftrightarrow \langle \mathfrak{F}^s, \delta, x \rangle \vDash \gamma_\circ \to \varphi^s$$

PROPOSITION 73. *Let $\mathfrak{F}$ be a bimodal generalized frame. $\mathfrak{F}^s \vDash \varphi^s$ iff $\mathfrak{F} \vDash \varphi$.*

For a formula χ and a set Δ put

(58) $$\chi \to \Delta := \{\chi \to \delta : \delta \in \Delta\}$$

We define **Sim** to be the logic of all simulating frames.

DEFINITION 74. Let L be a bimodal logic. Then L^s is the logic of all $\mathfrak{F}^s$, where $\mathfrak{F}$ is a general frame for L.

THEOREM 75. *Let L be a bimodal logic. Then*

(59) $$\Delta \vdash_L \varphi \quad \Leftrightarrow \quad \gamma_\circ \to \Delta^s \vdash_{L^s} \gamma_\circ \to \varphi^s$$

(60) $$\Delta \Vdash_L \varphi \quad \Leftrightarrow \quad \gamma_\circ \to \Delta^s \Vdash_{L^s} \gamma_\circ \to \varphi^s$$

In particular, if $L = \mathbf{K}_2 \oplus \Delta$, $L^s = \mathbf{Sim} \oplus (\gamma_\circ \to \Delta^s)$.

The previous result shows that the bimodal consequence is reduced to the consequence relation based on the 'white points' of the simulating frame.

4.2 Algebraic Properties of the Simulation

Let $p : \mathfrak{F} \to \mathfrak{G}$ be a p-morphism. Define p^s by $p^s(x_\circ) := p(x)_\circ$ and $p^s(x_\bullet) := p(x)_\bullet$, and $p^s(*) := *$. It is easy to see that p^s is a p-morphism from $\mathfrak{F}^s$ to $\mathfrak{G}^s$. Conversely, let $q : \mathfrak{F}^s \to \mathfrak{G}^s$ be a p-morphism. Then $q(*) = *$, $q[F_\circ] \subseteq G_\circ$ and $q[F_\bullet] \subseteq G_\bullet$, since all sets are definable by constant formulae. Next, if $q(x_\circ) = y_\circ$ then also $q(x_\bullet) = y_\bullet$, so that q is completely defined by its action on $F_\circ$. Moreover, $q = p^s$ for some p-morphism $p : \mathfrak{F} \to \mathfrak{G}$. So, the simulation is faithful with respect to embeddings and contractions.

Notice that $(\mathfrak{F} \oplus \mathfrak{G})^s$ is not isomorphic to $\mathfrak{F}^s \oplus \mathfrak{G}^s$ (the former is connected, the latter is not). However, the two are not so different. Basically, the latter has two points satisfying $\boxminus\bot$, the former only one. Thus only the former can be a simulation frame.

LEMMA 76. $(\bigoplus_{i \in I} \mathfrak{F}_i)^s$ *is a contraction of* $\bigoplus_{i \in I} \mathfrak{F}^s$*. The contraction is the one which collapses all points satisfying $\boxminus\bot$ into one.*

The construction can be remodeled algebraically. Let $\mathfrak{A}$ be a bimodal algebra.

(61) $$A^s := A \times A \times \{0, 1\}$$

(62) $$\diamondminus \langle a, b, c \rangle := \begin{cases} \langle \Diamond a \cup b, \blacklozenge b \cup a, 0 \rangle & \text{if } c = 0 \\ \langle A, \blacklozenge b \cup a, 0 \rangle & \text{if } c = 1 \end{cases}$$

$\mathfrak{A}^s$ is the simulating algebra for $\mathfrak{A}$. It is easy to verify that if $\mathfrak{A}$ is the algebra of subsets of $\mathfrak{F}$, $\mathfrak{A}^s$ is the algebra of subsets of $\mathfrak{F}^s$, and conversely. If $h : \mathfrak{A} \to \mathfrak{B}$ is a homomorphism, so is $h^s : \mathfrak{A}^s \to \mathfrak{B}^s$. Moreover, if $q : \mathfrak{A}^s \to \mathfrak{B}^s$ is a homomorphism, then $q = p^s$ for some

$p : \mathfrak{A} \to \mathfrak{B}$. So we have an isomorphism between the category of bimodal algebras and the category of simulation algebras.

We are interested in the varieties generated by simulation algebras. $\mathcal{V}(\mathbf{Sim})$ is the variety generated by all simulation algebras. It is easier to look at the frames. Take a nonempty generated subframe $\mathfrak{G}$ of $\mathfrak{F}^s$. It is easy to see that it must be of the form $\mathfrak{H}^s$. Simply take $H := G \cap F_\circ$ (notice that we do allow frames to be empty). However, the empty subframe is not of that form. So, with the exception of the empty frame every subframe of $\mathfrak{F}^s$ is a simulation frame. It follows that $\mathrm{Cg}(\mathfrak{A}^s) \cong \mathrm{Cg}(\mathfrak{A}) + 1$, where the latter denotes the addition of a new top element to $\mathrm{Cg}(\mathfrak{A})$.

PROPOSITION 77. *$\mathfrak{A}$ is subdirectly irreducible iff $\mathfrak{A}^s$ is.*

Next, let $p : \mathfrak{F}^s \to \mathfrak{G}$ be a contraction. It is easy to see that $p(x_\circ) = p(y_\bullet)$ cannot hold; also, $p(x_\circ) \neq p(*) \neq p(x_\bullet)$. Moreover, $p(x_\circ) \leqslant p(y_\bullet)$ iff $x = y$ iff $p(y_\bullet) \leqslant p(x_\circ)$; and if $p(x_\circ) = p(y_\circ)$ then $p(x_\bullet) = p(y_\bullet)$, and conversely. So, $\mathfrak{G} = \mathfrak{H}^s$ for some $\mathfrak{H}$. It follows that $\mathrm{Sub}(\mathfrak{A}^s) \cong \mathrm{Sub}(\mathfrak{A})$. Finally, we have noticed that $(\prod_{i\in I} \mathfrak{A}_i)^s \in \mathsf{S}(\prod_{i\in I} \mathfrak{A}_i^s)$.

Now, if $\mathcal{K}$ is a class of bimodal algebras, denote by $\mathcal{K}^\sigma := \{\mathfrak{A}^s : \mathfrak{A} \in \mathcal{K}\}$. Also, denote by $\mathcal{K}_{si}$ the class of subdirectly irreducible members of $\mathcal{K}$.

PROPOSITION 78. *If $\mathcal{V}$ is a variety of bimodal algebras, $(\mathcal{V}_{si})^\sigma = (\mathcal{V}^\sigma)_{si}$.*

Now let $\mathcal{V}^s$ be the variety generated by $\mathcal{V}^\sigma$. For any variety generated by simulation algebras, the subdirectly irreducible members are simulation algebras. Hence, any subvariety of $\mathcal{V}(\mathbf{Sim})$, with the exception of the trivial variety, is of the form $\mathcal{V}^s$.

THEOREM 79. *The map $\mathcal{V} \mapsto \mathcal{V}^s$ is an isomorphism from the lattice of varieties of bimodal algebras onto the lattice of nontrivial subvarieties of $\mathcal{V}(\mathbf{Sim})$.*

We now turn to the axiomatisation of the simulations. Put

$$
\begin{aligned}
&*(x) := \neg(\exists y)(x \leqslant y) \\
(63)\qquad &\circ(x) := (\exists y)(x \leqslant y \wedge *(y)) \\
&\bullet(x) := \neg * (x) \wedge \neg \circ (x)
\end{aligned}
$$

A monomodal frame is a simulation frame iff it satisfies the following elementary formulae (here, $\exists!$ is short for: 'there exists exactly one'):

(64a) $\quad (\forall x)(\circ(x) \to (\exists! y)(x \leqslant y \wedge \bullet(y)))$

(64b) $\quad (\forall x)(\bullet(x) \to (\exists! y)(x \leqslant y \wedge \circ(y)))$

(64c) $\quad (\exists! x)(*(x))$

(64a) and (64b) are modally definable, but (64c) is not. It turns out that the class of **Sim**-frames is the class of frames satisfying (64a) and (64b). The axiomatisation can be derived from the correspondence between first-order and modal formulae. Moreover, **Sim** is R-persistent. Hence, any logic containing **Sim** is complete with respect to simulation frames.

4.3 Unsimulation

Let $\mathfrak{F}$ be a **Sim**-frame. Then F° is the set of points satisfying $\circ(x)$, and $F^\bullet$ the set of points satisfying $\bullet(x)$. If $x \in F^\circ$ let $x^\dagger$ be the unique successor in $F^\bullet$, and if $x \in F^\bullet$ then

let $x^\dagger$ be the unique successor in F°. Now put $\mathfrak{F}_s := \langle F_s, \lhd, \blacktriangleleft, \mathbb{F}_s\rangle$, where

$$(65)\qquad \begin{array}{rl} F_s & := F^\circ \\ \lhd & := \leqslant \cap F_s^2 \\ \blacktriangleleft & := \{\langle x^\dagger, y^\dagger\rangle : x \leqslant y, x, y \in F_\bullet\} \\ \mathbb{F}_s & := \{a \in \mathbb{F} : a \subseteq F_s\} \end{array}$$

For the following theorem notice that $(\mathfrak{M}_s)^s$ always is connected, by construction.

PROPOSITION 80. *Let $\mathfrak{M}$ be a monomodal connected* **Sim***-frame, $\mathfrak{B}$ a bimodal frame. Then $\mathfrak{M} \cong (\mathfrak{M}_s)^s$ and $\mathfrak{B} \cong (\mathfrak{B}^s)_s$.*

For each variable p we introduce three variables $p_\circ$, $p_\bullet$, p_*. We call the new set the **extended set of variables**. Think of $p_\circ$ as 'p is true at the region of $\gamma_\circ$ worlds'; $p_\bullet$ as 'is true at the region of $\gamma_\bullet$ worlds' and p_* as 'is true at the region of γ_* worlds'. For formulae χ we define the formulae $\chi_\circ$, $\chi_\bullet$ and χ_* by mutual recursion (and think of them as interpreted in the same way as the new variables).

$$(66)\qquad \begin{array}{lll l} (\neg\varphi)_\alpha & := & \neg\varphi_\alpha & \alpha \in \{\bullet, \circ, *\} \\ (\varphi \wedge \chi)_\alpha & := & \varphi_\alpha \wedge \chi_\alpha & \alpha \in \{\bullet, \circ, *\} \\ (\Diamond\varphi)_\circ & := & \varphi_\bullet \vee \varphi_* \vee \Diamond\varphi_\circ & \\ (\Diamond\varphi)_\bullet & := & \varphi_\circ \vee \blacklozenge\varphi_\bullet & \\ (\Diamond\varphi)_* & := & \bot & \end{array}$$

Let β be a valuation on $\mathfrak{F}^s$. Define β_s on $\mathfrak{F}$ so that for all variables p (assuming $x_* = *$ and $\alpha \in \{\circ, \bullet, *\}$):

$$(67)\qquad \langle \mathfrak{F}, \beta_s, x\rangle \vDash p_\alpha \quad \Leftrightarrow \quad \langle \mathfrak{F}^s, \beta, x_\alpha\rangle \vDash p$$

Then it is established by induction on the formulae that

$$(68)\qquad \begin{array}{rl} & \langle \mathfrak{F}^s, \beta, x_\alpha\rangle \vDash \varphi \\ \Leftrightarrow & \langle \mathfrak{F}, \beta_s, x\rangle \vDash \varphi_\alpha \\ \Leftrightarrow & \langle \mathfrak{F}^s, \beta, x_\alpha\rangle \vDash \varphi^s_\circ(p/p_\circ, \Diamond_\bullet p/p_\bullet, \Diamond_* p/p_*) \end{array}$$

As usual, $\Diamond_\bullet\varphi := \neg \boxminus_\bullet \neg\varphi$ and similarly for $\Diamond_*$, where $\boxminus_\bullet$ and $\boxminus_*$ are as defined in Equation (54). Now, every valuation on $\mathfrak{F}$ of the extended set of variables is of the form β_s for some valuation β on $\mathfrak{F}^s$. Thus we obtain $\langle \mathfrak{F}^s, x_\alpha\rangle \vDash \varphi$ iff $\langle \mathfrak{F}, x\rangle \vDash \varphi_\alpha$, for every $\alpha \in \{\circ, \bullet, *\}$. Finally, this gives

$$(69)\qquad \mathfrak{F}^s \vDash \varphi \quad \Leftrightarrow \quad \mathfrak{F} \vDash \varphi_\circ \wedge \varphi_\bullet \wedge \varphi_*$$

Therefore, let

$$(70)\qquad \varphi_s := \varphi_\circ \wedge \varphi_\bullet \wedge \varphi_*$$

THEOREM 81. *Let $L = \mathbf{Sim} \oplus \Delta$ be consistent. Put $L_s := \mathbf{K}_2 \oplus \Delta_s$. Then $(L_s)^s = L$. Additionally,*

$$(71)\qquad \Delta \vdash_L \varphi \quad \Leftrightarrow \quad \Delta_s \vdash_{L_s} \varphi_s \qquad \textit{and} \qquad \Delta \Vdash_L \varphi \quad \Leftrightarrow \quad \Delta_s \Vdash_{L_s} \varphi_s$$

PROPOSITION 82. $\mathrm{Th}\,\mathfrak{F}^s = \mathbf{Sim} \oplus (\mathrm{Th}\,\mathfrak{F})^s$.

4.4 The Main Theorem

Let StSim be the category of differentiated monomodal **Sim**-frames, with γ_* containing a single point. The morphisms are the p-morphisms. Let Dif_2 be the category of differentiated bimodal frames with p-morphisms as maps.

THEOREM 83. StSim *and* Dif_2 *are naturally equivalent. The map* $(-)_s$ *is a functor from* StSim *to* Dif_2*,* $(-)^s$ *a functor from* Dif_2 *to* StSim*. Moreover, there is a natural transformation from the identity on* Dif_2 *to* $((-)^s)_s$ *and a natural transformation from the identity on* StSim *to* $((-)_s)^s$*.*

From Theorems 75 and 81 follows that $L \mapsto L^s$ preserves and reflects finite and recursive axiomatisability.

Next, if $L = \mathrm{Th}\,\mathcal{K}$ then $L^s = \mathrm{Th}\,\mathcal{K}^s$ and conversely. It follows that completeness, finite model property, tabularity are preserved and reflected.

Second, suppose that L is Df-persistent. Let $\mathfrak{M}$ be a differentiated monomodal frame for L^s. Then $\mathfrak{M}_s$ is differentiated and $\mathfrak{M} \cong (\mathfrak{M}_s)^s$, which is therefore differentiated. It follows that $\mathfrak{M}_s \vDash L$, so that the underlying Kripke-frame $(\mathfrak{M}_s)_\sharp$ is an L^s-frame. Since $(\mathfrak{M}_s)_\sharp \cong (\mathfrak{M}_\sharp)_s$, we have $\mathfrak{M}_\sharp \vDash L^s$. Similarly for R-persistence.

Now we turn to interpolation. Using the algebraic characterisation of interpolation (Theorem 93) the preservation and reflection of interpolation is actually straightforward to show. There are also direct ways. Suppose that the bimodal logic L has interpolation. Now let $\varphi \vdash_{L^s} \psi$. Putting together Equations (56) and (68) we get that for every ρ,

(72) $$\gamma_\circ \vdash_{L^s} \rho \leftrightarrow \sigma(\rho_s)^s$$

where $\sigma : p_\circ \mapsto p, p_\bullet \mapsto \diamondsuit_\bullet p, p_* \mapsto \diamondsuit_* p$. So $\gamma_\circ \to (\varphi_s)^s \vdash_{L^s} \gamma_\circ \to (\psi_s)^s$, and so by Theorem 75, and the fact that $L = (L^s)_s$ we get $\varphi_s \vdash_L \psi_s$. There exists a formula χ such that $\mathrm{var}(\chi) \subseteq \mathrm{var}(\varphi_s) \cap \mathrm{var}(\psi_s)$ and $\varphi_s \vdash_L \chi \vdash_L \psi_s$. Now, χ is in the variables $p_\circ, p_\bullet, p_*$ for $p \in \mathrm{var}(\varphi) \cap \mathrm{var}(\psi)$, and this applies as well to χ^s. Furthermore,

(73) $$\gamma_\circ \to (\varphi_s)^s \vdash_{L^s} \chi^s \vdash_{L^s} \gamma_\circ \to (\psi_s)^s$$

so that

(74) $$\gamma_\circ \to \varphi \equiv \gamma_\circ \to \sigma((\varphi_s)^s) \vdash_{L^s} \sigma(\chi^s) \vdash_{L^s} \gamma_\circ \to \sigma((\psi_s)^s) \equiv \gamma_\circ \to \sigma((\psi_s)^s)$$

Put $\chi^\circ := \gamma_\circ \to \sigma(\chi^s)$. Likewise formulae χ' and χ'' can be found such that

(75) $$\gamma_\bullet \to \varphi \vdash_{L^s} \gamma_\bullet \to \sigma(\chi'^s) \vdash_{L^s} \gamma_\bullet \to \psi$$

(76) $$\gamma_* \to \varphi \vdash_{L^s} \gamma_* \to \sigma(\chi''^s) \vdash_{L^s} \gamma_* \to \psi$$

Put $\chi^\bullet := \gamma_\bullet \to \diamondsuit_\circ \sigma(\chi'^s)$, and $\chi^* := \gamma_* \to \sigma(\chi'')$. Then $\chi^\circ \wedge \chi^\bullet \wedge \chi^*$ is the desired interpolant. The proof works analogously for global interpolation and transfer of local and global Halldén-completeness.

Now look at Sahlqvist formulae. By a theorem of [39] a modal logic is Sahlqvist iff it can be axiomatized by formulae of the form $\varphi \to \psi$, where ψ and φ is composed from compound modalities using only $\wedge$, $\vee$ and diamonds. (Compound modalities are **strongly positive** formulae.) From this it follows immediately that if $\varphi \to \psi$ is Sahlqvist, so is $(\varphi \to \psi)^s = \varphi^s \to \psi^s$. (The original formulation allows a prefix of boxes but this does not

define a larger class of logics.) The converse is similar. We have $(\varphi \to \psi)_s = \varphi_s \to \psi_s$, and the unsimulation translates boxes into boxes and diamonds into diamonds. Finally, note that simulation and unsimulation commute with ultraproducts.

THEOREM 84 (Kracht & Wolter). *The map $L \mapsto L^s$ is an isomorphism from the locale of normal bimodal logics onto the interval $[\mathbf{Sim}, \mathrm{Th}\,\boxed{\bullet}]$ in the locale of normal monomodal logics. Moreover, the following properties of logics are invariant under this map:*

1. *decidability,*
2. *elementarity, Df-persistence, R-persistence, being Sahlqvist,*
3. *finite model property, completeness, compactness,*
4. *local and global interpolation.*

Halldén-completeness is actually *not* preserved under simulation. For example, the logic $\mathbf{D} \otimes \mathbf{D}$ is Halldén-complete (being the fusion of two Halldén-complete logics). However, its simulation has more than two constants, so it cannot be Halldén-complete (see below, Theorem 99).

In general, for $\kappa \in \omega$ there is a similar isomorphism from $\mathrm{NExt}(\mathbf{K}_\kappa)$ onto an interval $[\mathbf{Sim}_\kappa, \mathrm{Th}(\mathfrak{Ch}_{\kappa-1})]$, where $\mathbf{Sim}_\kappa$ is the simulating logic for κ-modal frames, and $\mathfrak{Ch}_{\kappa-1} = \langle\{0, 1, \ldots, \kappa-1\}, \prec\rangle$ $i \prec j$ iff $j = i+1$ if the chain of $\kappa - 1$ many points. The underlying set is $F \times \{0, 1, \ldots, \kappa-1\}$ plus the points of the chain, and we put $\langle x, i\rangle \leqslant \langle y, j\rangle$ iff (a) $x = y$ or (b) $i = j$ and $x \lhd_i y$. Additionally, every point $\langle x, i\rangle$ sees the point $i - 1$. Finally, $i \leqslant j$ iff $i \prec j$. All aforementioned properties are invariant under this simulation. For countable κ, [38] describes an embedding of $\mathrm{NExt}\,\mathbf{K}_\kappa$ into (not necessarily onto) an interval in $\mathrm{NExt}\,\mathbf{K}_1$.

4.5 Simulating Polyadic and Nonstandard Operators by Monadic Operators

Assume that L is a complete logic. Now add two modal operators $\boxright$ and $\boxleft$, together with the following axioms: **G3** for $\boxleft$, **K4.3** for $\boxright$, $p \to \boxright\diamondleft p$, $p \to \boxleft\diamondright p$ and $(\boxright p \wedge \boxleft p \wedge p) \to \boxplus p$ for every basic modality $\boxplus$. Thus, $\boxright$ and $\boxleft$ are tense duals, the relation for $\boxright$ is a well-order, and if $x\ R(\boxplus)\ y$ for any compound modality $\boxplus$ then either $x = y$ or $x\ R(\boxright)\ y$ or $y\ R(\boxright)\ x$. Call this logic L^w. The difference operator of [16, 17] and the universal modality are now definable on all connected frames by

(77) $$[\neq]\chi := \boxright\chi \wedge \boxleft\chi \qquad [u]\chi := \chi \wedge [\neq]\chi$$

Also, [2] has introduced a logic using a special type of variables, called **nominals**, which must be interpreted by singleton sets. Logics that admit both standard variables and nominals are called **hybrid**, see Chapter 14 of this handbook. It turns out that with the difference modality the standard languages have the same expressive power as the hybrid ones. Consider a variable p. Put

(78) $$n(p) := \langle u\rangle(p \wedge [\neq]\neg p)$$

β satisfies $n(p)$ on a Kripke-frame iff the value of p is a singleton set. (In [29] the operator $Op := p \wedge [\neq]p$ is called the 'only' operator. It says that p is true 'only here'.) In absence of the difference modality, the nominals give extra expressive power. Consider the operator $\blacksquare$ defined by the following axiom:

$$n(p) \to (\blacklozenge p \leftrightarrow \neg\Diamond p) \tag{79}$$

Then a frame satisfies this axiom iff $\blacktriangleleft$ is the complement of $\lhd$ (the inaccessibility relation of [34]). Notice that the following holds.

THEOREM 85 (Gargov & Goranko). *A class of frames is definable using the language with nominals and the universal modality iff it is definable using the difference operator.*

Recall the characterisation of the first-order properties axiomatisable by means of Sahlqvist-formulae. Using the inaccessibility relation and the universal modality we can not only express unrestricted quantification (on connected frames) but also negative formulae. This means that all first-order conditions over binary relations are now expressible (over the logic of these structures with enough modal operators) in which an atomic formula contains at least one universally quantified variable whose quantifier is not in the scope of an existential. This last restriction can be circumvented through the introduction of new modalities (to mimic the Skolem functions) on condition that the variable depends only on one other variable. All these codings proceed by adding more operators, not more points. For example, **ZFC** without foundation can be so axiomatized, see [38]. The infinity axiom can be expressed much more succinctly than in that paper. Simply require

$$(\exists x)(\varnothing \in x \wedge (\forall y)(y \in x \to (\exists z)(y \in z \in x))) \tag{80}$$

The outer existential can be massaged away by introducing a constant, and the second existential can be dealt with using a Skolem function. Foundation of course is axiomatisable using the **G**-axioms for $\ni^+$. Even full class comprehension is axiomatisable.

If one is interested in simulating polyadic operators then it is not enough to just add relations (similarly if one wants to simulate predicate logic with a signature containing at least ternary relations symbols). One approach was outlined in [40]. A better one is presented in [28]. It is enough to look at the case of a single binary modal operator ∇. **Kripke-frames** are pairs $\langle F, R\rangle$ where R is a ternary relation. A **generalized frame** is a triple $\mathfrak{F} = \langle F, R, \mathbb{F}\rangle$ where $\mathbb{F} \subseteq \wp(F)$ is a field of sets closed under

$$\mu_R(A_1, A_2) := \{x \in F : (\exists v_1 \in A_1)(\exists v_2 \in A_2)R(x, v_1, v_2)\} \tag{81}$$

Satisfaction of a formula is defined as follows.

$$\langle \mathfrak{F}, \beta, x\rangle \vDash \nabla(\varphi_1, \varphi_2) \quad \Leftrightarrow \quad \begin{array}{r} \text{there are } v_1, v_2 \text{ such that } R(x, v_1, v_2) \\ \text{and } \langle \mathfrak{F}, \beta, v_1\rangle \vDash \varphi_1; \langle \mathfrak{F}, \beta, v_2\rangle \vDash \varphi_2 \end{array} \tag{82}$$

For a triple $\vec{x}$, let $\vec{x}_i$ be the ith component of $\vec{x}$. Given $\mathfrak{F}$, assume $F \cap R = \varnothing$ and put

$$\begin{array}{rl} F^\bullet & := F \cup R \\ R_i & := \{\langle \vec{x}, \vec{x}_i\rangle : \vec{x} \in R\} \\ S & := \{\langle \vec{x}_0, \vec{x}\rangle : \vec{x} \in R\} \\ \mathbb{F}^\bullet & := \{a \cup (R \cap \bigcup_{i<n} a_i \times b_i \times c_i) : a, a_i, b_i, c_i \in \mathbb{F}, n < \omega\} \\ \mathfrak{F}^\bullet & := \langle F^\bullet, S, R_0, R_1, R_2, \mathbb{F}^\bullet\rangle \end{array} \tag{83}$$

$\mathfrak{F}^\bullet$ is a general frame and $\mathbb{F}^\bullet$ is generated by $\mathbb{F}$. The set R is definable by $\mu := \langle R_0 \rangle \top$. The set F in $F^\bullet$ is called the set of **base points**. It too is definable. The simulation is now defined as follows.

(84)
$$\begin{aligned} p^\bullet &:= p \\ (\neg\varphi)^\bullet &:= \neg\varphi^\bullet \\ (\varphi_1 \wedge \varphi_2)^\bullet &:= \varphi_1^\bullet \wedge \varphi_2^\bullet \\ (\nabla(\varphi_1, \varphi_2))^\bullet &:= \langle S \rangle (\langle R_1 \rangle \varphi_1^\bullet \wedge \langle R_2 \rangle \varphi_2^\bullet) \end{aligned}$$

The translation $(\cdot)^\bullet$ preserves truth of formulae at base points. So we get

PROPOSITION 86. $\mathfrak{F} \vDash \varphi \quad \Leftrightarrow \quad \mathfrak{F}^\bullet \vDash \neg\mu \to \varphi^\bullet$.

We remark that $\mathfrak{F}^\bullet$ is differentiated (descriptive) iff $\mathfrak{F}$ is.

Unsimulation is less straightforward. Let $\langle M, S, R_0, R_1, R_2, \mathbb{M} \rangle$ be a monadic frame. Put

(85)
$$\begin{aligned} M_\bullet &:= \{x : x \vDash \neg\mu\} \\ T &:= \{\vec{x} : \exists v : v\, R_0\, \vec{x}_0, v\, R_1\, \vec{x}_1 \text{ and } v\, R_2\, \vec{x}_2\} \\ \mathbb{M}_\bullet &:= \{a \cap M_\bullet : a \in \mathbb{M}\} \end{aligned}$$

It is straightforward to check that for a ternary frame $\mathfrak{F} \cong (\mathfrak{F}^\bullet)_\bullet$.

Now let $\mathbf{Sim}^\bullet$ be the logic of general frames of the form $\mathfrak{F}^\bullet$. The simulation map sends a dyadic logic L to the logic

(86)
$$L^\bullet := \mathbf{Sim}^\bullet \oplus \{\neg\mu \to \varphi^\bullet : \varphi \in L\}$$

This map turns out to be a lattice homomorphism. It is injective but not surjective, unlike in the monadic case (in finite signature). A useful observation is this.

PROPOSITION 87. *Let L be a dyadic logic and $\mathcal{K}$ a class of $\mathbf{Sim}^\bullet$-frames. If $L^\bullet$ is complete with respect to $\mathcal{K}$ then L is complete with respect to $\mathcal{K}_\bullet$.*

Given an extension L of $\mathbf{Sim}^\bullet$, let $L_\bullet$ be the logic generated by all formulae valid on all unsimulations of descriptive L-frames.

PROPOSITION 88. *The following holds for a dyadic logic L and an extension M of $\mathbf{Sim}^\bullet$.*

① $M_\bullet = \{\varphi : \neg\mu \to \varphi^\bullet \in M\}$.

② $L \subseteq M_\bullet$ *iff* $L_\bullet \subseteq M$.

③ $(M_\bullet)^\bullet \subseteq M$.

④ $(L^\bullet)_\bullet = L$.

The following is shown in [28].

THEOREM 89 (Goguadze & Piazza & Venema). *The map $L \mapsto L^\bullet$ is a lattice homomorphism into the lattice* NExt($\mathbf{Sim}^\bullet$). *It preserves and reflects*

1. *finite and recursive axiomatisability,*

2. *completeness, finite model property, tabularity,*

3. canonicity,

4. first-order definability.

It preserves Sahlqvist axiomatisability and it reflects decidability.

5 INTERPOLATION

This section uses some algebraic notions that are either covered at the beginning of this chapter or in Chapter 6. Notice also the discussion on interpolation and fusion in Chapter 14, as well as simulations and fusion discussed in the previous section.

5.1 Algebraic Characterisation

DEFINITION 90. A modal logic L has **local interpolation** if $\vdash_L$ has interpolation; it has **global interpolation** if $\Vdash_L$ has interpolation.

Since $\vdash_L$ has a deduction theorem, local interpolation can also formulated as follows: if $\varphi \to \psi \in L$ there is a χ such that $\mathrm{var}(\chi) \subseteq \mathrm{var}(\varphi) \cap \mathrm{var}(\psi)$ and $\varphi \to \chi, \chi \to \psi \in L$. This property is also known as **Craig interpolation**. Notice that interpolation is a property of the consequence relation not of the logic.

PROPOSITION 91. *If L has local interpolation it also has global interpolation.*

Proof. Assume $\varphi \Vdash_L \psi$. Then for some compound modality $\boxplus$, $\boxplus\varphi \vdash_L \psi$. By assumption there is a χ in the joint variables such that $\boxplus\varphi \vdash_L \chi \vdash_L \psi$. It follows that $\varphi \Vdash_L \chi \Vdash_L \psi$. ❑

DEFINITION 92. A variety $\mathcal{V}$ of modal algebras has the **amalgamation property** if for every triple $\mathfrak{A}_0$, $\mathfrak{A}_1$ and $\mathfrak{A}_2$ from $\mathcal{V}$ and embeddings $\iota_1 : \mathfrak{A}_0 \rightarrowtail \mathfrak{A}_1$, $\iota_2 : \mathfrak{A}_0 \rightarrowtail \mathfrak{A}_2$ there is a $\mathfrak{B} \in \mathcal{V}$ and embeddings $\varepsilon_1 : \mathfrak{A}_1 \to \mathfrak{B}$, $\varepsilon : \mathfrak{A}_2 \to \mathfrak{B}$ such that $\varepsilon_1 \circ \iota_1 = \varepsilon_2 \circ \iota_2$. $\mathcal{V}$ has the **superamalgamation property** if in addition to the above for every $a_1 \in A_1$ and $a_2 \in A_2$: (a) if $\varepsilon_1(a_1) \leq \varepsilon_2(a_2)$ then there is a $c \in A_0$ such that $a_1 \leq \iota_1(c)$ and $\iota_2(c) \leq a_2$ and (b) if $\varepsilon_1(a_1) \geq \varepsilon_2(a_2)$ then there is a $c \in A_0$ such that $a_1 \geq \iota_1(c)$ and $\iota_2(c) \geq a_2$.

$$\begin{tikzcd}
& \mathfrak{A}_1 \arrow[dr, dotted, "\varepsilon_1"] & \\
\mathfrak{A}_0 \arrow[ur, "\iota_1"] \arrow[dr, "\iota_2"'] & & \mathfrak{A}_3 \\
& \mathfrak{A}_2 \arrow[ur, dotted, "\varepsilon_2"'] &
\end{tikzcd} \tag{87}$$

THEOREM 93 (Maksimova). *Let L be a modal logic.*

1. L has local interpolation iff $\mathcal{V}(L)$ has the superamalgamation property.

2. L has global interpolation iff $\mathcal{V}(L)$ has the amalgamation property.

We sketch a proof of the second claim. Suppose that L has global interpolation, and let $\mathfrak{A}_0$, $\mathfrak{A}_1$ and $\mathfrak{A}_2$ plus two embeddings be given. We define $\mathfrak{F}_i := \mathfrak{Fr}_L(A_i)$ and $\mathfrak{F}_3 := \mathfrak{Fr}_L(A_1 \cup A_2)$, where $\mathfrak{Fr}_L(X)$ denotes the algebra freely generated by X in the variety of L-algebras. The embeddings form a commuting square as in (87). The identity map induces a surjective homomorphism $\pi_i : \mathfrak{F}_i \twoheadrightarrow \mathfrak{A}_i$. For $i = 1, 2$ put $T_i := \{\varphi : \pi_i(\varphi) = 1\}$. Let $T := \{\chi : T_1 \cup T_2 \Vdash_L \chi\}$. Then the following holds for $\varphi \in F_1$, $\psi \in F_2$:

$$(88) \qquad T \Vdash_L \varphi \to \psi \quad \Leftrightarrow \quad (\exists \chi \in F_0)(\varphi \to \chi \in T_1 \text{ and } \chi \to \psi \in T_2)$$

For if $T \Vdash_L \varphi \to \psi$ then there are finite $\Gamma_i \subseteq T_i$ and a compound modality such that $\boxplus\Gamma_1; \boxplus\Gamma_2 \vdash_L \varphi \to \psi$, giving $\Gamma_1; \varphi \Vdash_L \boxplus\Gamma_2 \to \psi$. There is an interpolant $\chi \in F_0$, from which we deduce $\varphi \to \chi \in T_1$ and $\chi \to \psi \in T_2$.

Put $\varphi \,\Theta\, \psi$ iff $T \Vdash_L \varphi \leftrightarrow \psi$. This is a congruence on $\mathfrak{F}_3$ and we put $\mathfrak{A}_3 := \mathfrak{F}_3/\Theta$. Using the above property it is shown that for $i = 1, 2$ and $\varphi \in F_i$: $\varphi \,\Theta\, \top$ implies $\varphi \in T_i$. So, the natural map $\mathfrak{F}_i \twoheadrightarrow \mathfrak{A}_3$ factors through π_i, giving a map $\varepsilon_i : \mathfrak{A}_i \to \mathfrak{A}_3$ with the desired properties.

Conversely, assume that $\mathcal{V}(L)$ has the amalgamation property. Let $\varphi = \varphi(\vec{p}, \vec{r})$ and $\psi = \psi(\vec{r}, \vec{q})$ be given such that no global interpolant exists. We shall show that $\varphi \nVdash_L \psi$. Let $\mathfrak{F}_0 := \mathfrak{Fr}_L(\vec{r})$, $\mathfrak{F}_1 := \mathfrak{Fr}_L(\vec{p}, \vec{r})$, $\mathfrak{F}_2 := \mathfrak{Fr}_L(\vec{q}, \vec{r})$, and $\mathfrak{F}_3 := \mathfrak{Fr}_L(\vec{p}, \vec{q}, \vec{r})$. Let O_1 be an open filter in $\mathfrak{F}_1$ containing φ and O_2 an open filter containing $\neg\psi$ and $O_1 \cap F_0$. Then $O_3 := O_2 \cap F_0 = O_1 \cap F_0$. Let Θ_i be the congruence associated with O_i, $\mathfrak{A}_i := \mathfrak{F}_i/\Theta_i$. Then for all $\chi, \chi' \in F_0$, $\chi \,\Theta_1\, \chi'$ iff $\chi \,\Theta_2\, \chi'$. So we have embeddings $\iota_i : \mathfrak{A}_0 \rightarrowtail \mathfrak{A}_i$. Now we get an algebra $\mathfrak{B}$ and maps $\varepsilon_i : \mathfrak{F}_i \to \mathfrak{B}$ such that $\varepsilon_1 \circ \iota_1 = \varepsilon_2 \circ \iota_2$. Define v by $v(p) := \varepsilon_1([p]\Theta_1)$ if $p \in \mathrm{var}(\varphi)$ and $v(p) := \varepsilon_2([p]\Theta_2)$ if $p \in \mathrm{var}(\psi)$. Since $\varphi \in O_1$, $v(\boxplus\varphi) = 1$ for all compound modalities. Since $\overline{v}(\neg\psi) \neq 1$, $\varphi \nVdash_L \psi$.

We remark here that the proof established that the category of L-algebras has pushouts for monomorphisms.

THEOREM 94 (Maksimova). *There are exactly seven consistent logics containing* **Grz** *which have interpolation. There are at most 50 consistent logics containing* **S4** *which have global interpolation and at most 37 logics having interpolation.*

The first result is from [42], the second from [43]. We sketch a proof, restricting our attention to global interpolation. The first step is to notice that if a modal logic L has interpolation, then so does the intermediate logic determined by this class of frames (under the Gödel translation). Notice the following. For a (general) frame $\mathfrak{F}$, define the skeleton $S(\mathfrak{F})$ by reducing every cluster to size 1. If $L \supseteq$ **S4** is determined by the class $\mathcal{K}$ of general frames, then the intermediate logic associated with it is determined by the class $\{S(\mathfrak{F}) : \mathfrak{F} \in \mathcal{K}\}$. It is not hard to see that if $\mathcal{K}$ has amalgamation then so does $S(\mathcal{K})$. Therefore, the first step is to characterize the intermediate logics which have interpolation.

It is best to use the dual characterisation in terms of frames: a necessary condition for a logic to have global interpolation is that if $p_1 : \mathfrak{F}_1 \to \mathfrak{F}_0$ and $p_2 : \mathfrak{F}_2 \to \mathfrak{F}_0$ are surjective p-morphisms of L-frames there is an L-frame $\mathfrak{G}$ and p-morphisms $q_1 : \mathfrak{G} \to \mathfrak{F}_1$ and $q_2 : \mathfrak{G} \to \mathfrak{F}_2$ such that $p_1 \circ q_1 = p_2 \circ q_2$. Call $\mathfrak{G}$ a **fibred product** of the $\mathfrak{F}_i$. Now suppose that the logic contains only frames of depth $\leq n$, where $n > 2$, and that it contains the chain of length n, which is the frame $\mathfrak{L}_n = \langle \{x_i : i < n\}, \lhd \rangle$ with $x_i \lhd x_j$ iff $i \leq j$. Now define two maps: $p_1(x_i) = x_i$ if $i < n-1$ and $p_1(x_{n-1}) = x_{n-2}$; $p_2(x_i) = x_{i-1}$ if $i > 0$, and $p_2(0) = x_0$. It is easily seen that there is no fibred product $\mathfrak{G}$ of depth n; there only is one of depth $n + 1$. This observation leads to the following result.

Table 1. Intermediate Logics with Interpolation

Name	Axiomatisation	Characteristic Frames
Int		all **Grz**-frames
LC	$(p \to q) \vee (q \to p)$	all linear frames
$\mathbf{BD}_2$	$p \vee (p \to (q \vee \neg q))$	all frames of depth 2
KC	$\neg p \vee \neg\neg p$	all confluent frames
$\mathbf{BD}_2.\mathbf{BW}_2$	$p \vee (p \to (q \vee \neg q))$, $(p \to q) \vee (q \to p) \vee (p \leftrightarrow \neg q)$	$\mathfrak{B}_2$
$\mathbf{LC}_2$	$\neg p \vee \neg\neg p, p \vee (p \to (q \vee \neg q))$	the two element chain
PC	$p \vee \neg p$	the one element frame
Inc	p	no frames

LEMMA 95. *Let $L \supseteq$ **S4** have global interpolation. If $\mathfrak{L}_3$ is an L-frame, then every $\mathfrak{L}_n$, $n \in \omega$, is an L-frame.*

Notice that every frame of depth n can be mapped onto $\mathfrak{L}_n$, so that if L contains a frame of depth at least 3, it has frames of any given depth. This can be generalized. Let $\mathfrak{F} = \langle F, \lhd \rangle$ and $\mathfrak{G} = \langle G, \blacktriangleleft \rangle$ be frames with $F \cap G = \varnothing$; then let $\mathfrak{F} \oslash \mathfrak{G} := \langle F \cup G, \prec \rangle$, where $x \prec y$ iff (a) $x, y \in F$ and $x \lhd y$ or (b) $x \in F$, $y \in G$ or (c) $x, y \in \mathfrak{G}$ and $x \blacktriangleleft y$. Further, let $\circ$ denote the one-element reflexive frame.

LEMMA 96. *Let $L \supseteq$ **S4** have global interpolation. If $\mathfrak{F} \oslash \circ \oslash \circ$ is an L-frame, then so is $\mathfrak{F}(\oslash\circ)^n \oslash \circ$ for every $n \in \omega$.*

Notice that for every frame, it is possible to collapse the points of depth j into a single point, if one is doing that for all $j < m$, m given. This means that the previous theorem restricts the set of logics with interpolation enormously.

Now we turn to branching. Let $\mathfrak{B}_n := \langle \{x_i : i < n+1\}, \lhd \rangle$, where $x_0 \lhd x_i$ for every $i < n+1$, and $x_i \lhd x_j$ iff $i = j$ when $i > 0$. Similarly, it is established that if L has interpolation and contains $\mathfrak{B}_3$ then it contains all $\mathfrak{B}_n$. A related result is that if L has a frame in which a node branches into 3 immediate successors, then it has unbounded branching. Finally, let $\mathfrak{K}_n := \langle \{y_i : i < n\} \cup \{x, z\}, \lhd \rangle$, where for all $i < n$ we have (a) $x \lhd x$, (b) $x \lhd z$, (c) $x \lhd y_i$, (d) for all $i, j < n$: $y_i \lhd y_i$, (e) for all $i < n$: $y_i \lhd z$ and (f) $z \lhd z$; and no other relations hold. First consider the p-morphisms $q_0, q_1 : \mathfrak{K}_2 \to \mathfrak{L}_3$ defined by $q_0(x) \neq q_0(y_1) \neq q_0(y_0) = q_0(z)$ and $q_1(x) \neq q_1(y_0) \neq q_1(y_1) = q_1(z)$. The fibred product is the frame $\mathfrak{K}_3$. Iterating this argument gives us that all $\mathfrak{K}_n$ must be L-frames. Next consider the p-morphisms $p_0, p_1 : \mathfrak{K}_2 \to \mathfrak{L}_2$ defined by $p_0(y_0) = p_0(y_1) = p_0(x) \neq p_0(z)$ and $p_1(x) \neq p_1(y_0) = p_1(y_1) = p_1(z)$. The fibred product is a frame of depth 3 which has the structure $\circ \oslash (\circ \oplus \circ \oplus \circ) \oslash (\circ \oplus \circ \oplus \circ) \oslash \circ$. This means that as soon as $\mathfrak{K}_2$ is an L-frame, more and more frames can be shown to be L-frames, so that we can eventually conclude that $L =$ **Int**.

These results shall suffice to motivate the result that at most 7 consistent intermediate logics have interpolation. They are listed in Table 1. Now we turn to **S4**. We repeat the strategy with the clusters. Let $\mathfrak{Cl}_n = \langle \{x_i : i < n\}, \lhd \rangle$ with $x_i \lhd x_j$ for all $i, j < n$ be the n-element cluster. It can be shown that if a logic has a frame with a final 3-element cluster, then it has final clusters of arbitrary size. Similarly for nonfinal clusters. Now let $\mathcal{K}$ be a class of **Grz**-frames that has superamalgamation. Then we can derive the

following nine possibilities for classes of **S4**-frames: (a) allow the final clusters to be of size 1, 2 or limitless, (b) allow the nonfinal clusters to be of size 1, 2 or limitless. Applied to any of the 7 consistent logics we derive a maximum of 63 combinations of logics that have global interpolation. Some of these combinations are meaningless (for example, allowing the nonfinal clusters for **PC** to be proper), so that the list can be further reduced.

The notion of Halldén-completeness also splits into a local and a global version.

DEFINITION 97. A modal logic L is **locally Halldén-complete** if $\vdash_L$ is Halldén-complete; L is **globally Halldén-complete** if $\Vdash_L$ is Halldén-complete.

Global Halldén-completeness is also called the **pseudo relevance property**. The following is clear: if L is locally or globally Halldén-complete, it has up to equivalence at most two constants. For let φ be constant. $\varphi \vdash_L \varphi$, which by Halldén-completeness yields that φ is either inconsistent or a tautology. L has at most two constants iff $\Diamond\top \in L$ (iff $L \supseteq \mathbf{K.D}$) or $\Box\bot \in L$ (iff L is inconsistent or the logic of the one point irreflexive frame).

DEFINITION 98. A variety $\mathcal{V}$ of modal algebras has **fusion** if for every pair $\mathfrak{A}_1, \mathfrak{A}_2 \in \mathcal{V}$ there is a $\mathfrak{B} \in \mathcal{V}$ and embeddings $\varepsilon_1 : \mathfrak{A}_1 \to \mathfrak{B}$, $\varepsilon_2 : \mathfrak{A}_2 \to \mathfrak{B}$. $\mathcal{V}$ has **superfusion** if in addition to the above for every $a \in A_1 - \{0\}$ and every $b \in A_2 - \{1\}$ we have $\varepsilon_1(a) \nleq \varepsilon_2(b)$.

It is not hard to see that $\mathcal{V}(L)$ has fusion iff it has finite coproducts.

THEOREM 99 (Maksimova). *Let L be a modal logic.*

1. *L is locally Halldén-complete iff $\mathcal{V}$ has superfusion and the zero-generated algebras has at most two elements.*

2. *L is globally Halldén-complete iff $\mathcal{V}$ has fusion and the zero-generated algebras has at most two elements.*

The proof is essentially the same as in the case of interpolation. For the algebra $\mathfrak{Fr}_L(0)$ consists of two elements. For given two algebras $\mathfrak{A}_1$ and $\mathfrak{A}_2$, if they are nontrivial there are maps $\iota_i : \mathfrak{Fr}_L(0) \rightarrowtail \mathfrak{A}_i$. Using the same proof we obtain an algebra $\mathfrak{B}$ and embeddings $\varepsilon_i : \mathfrak{A}_i \rightarrowtail \mathfrak{B}$ such that $\varepsilon_1 \circ \iota_1 = \varepsilon_2 \circ \iota_2$.

5.2 Proving Interpolation

Besides the algebraic characterisation there are at least two other methods to prove interpolation. The first is based on tableau calculi and is basically due to [52]. This method can only be used if the tableau rules meet certain structural criteria. We show here only the case of **K**. Given a tableau calculus for L we do the following. Suppose that $\varphi \vdash_L \psi$. Then $\varphi; \neg\psi$ is L-inconsistent. So it has a closing tableau. We label the formulae in the tableau ^{a} if they derive from the formula φ, and ^{c} if they derive from $\neg\psi$. (If χ is a subformula of both, we create two copies of χ, namely χ^a and χ^c.) From the closing tableau we construct *two* closing tableaux, one for $\varphi^a; (\neg\chi)^c$, and one for $\chi^a; \neg\psi^c$. Moreover, χ will be based on the common variables of φ and ψ. This gives $\varphi \vdash_L \chi$ and

$\chi \vdash_L \psi$. Here is the calculus.

$$
(89) \qquad
\begin{array}{llll}
(w) & \dfrac{\Delta; \Sigma}{\Delta} & (\neg E) & \dfrac{\Delta; \neg\neg\varphi}{\Delta; \varphi} \\
(\wedge E) & \dfrac{\Delta; \varphi \wedge \psi}{\Delta; \varphi; \psi} & (\vee E) & \dfrac{\Delta; \neg(\varphi \wedge \psi)}{\Delta; \neg\varphi \mid \Delta; \neg\psi} \\
(\Box E) & \dfrac{\Box\Delta; \neg\Box\varphi}{\Delta; \neg\varphi} & &
\end{array}
$$

A **K-tableau** is a tree $\mathcal{C}$ constructed according to these rules. $\mathcal{C}$ **closes** if all leaves are of the form $\bot$ or $p; \neg p$, p a variable. Suppose $\mathcal{C}$ closes. The construction of the interpolant is bottom up. We show: *If $\Delta^a; \Sigma^c$ has a closed tableau there is a formula χ such that $\Delta^a; (\neg\chi)^c$ and $\Sigma^c; \chi^a$ both have a closed tableau.* The proof is by induction on the length of a closing tableau for $\Delta^a; \Sigma^c$. χ will be an interpolant for the sequent $\Delta \vdash \neg\Sigma$, where $\neg\Sigma$ is read disjunctively.

There are in total six cases for the leaves. (1) $p^a; (\neg p)^a$, (2) $p^a; (\neg p)^c$, (3) $(\neg p)^a; p^c$, (4) $p^c; (\neg p)^c$, (5) $\bot^a$ and (6) $\bot^c$. In Case (1), choose $\chi := \bot$, and the first tableau will end in $p^a; (\neg p)^a; (\neg\bot)^c$, the second in $\bot^a$. In Case (2), choose $\chi := p$. The first tableau will consist in $p^a; (\neg p)^c$, the second in $(\neg p)^c; p^a$. The Cases (3) and (4) are dual to (1) and (2). In Case (5), let $\chi := \bot$, in Case (6) $\chi := \neg\bot$. Now, suppose that the last step has been an application of $(\Box E)$. With labeling, the step is one of the following.

$$
(90) \qquad \frac{(\Box\Delta)^a; (\Box\Sigma)^c; (\neg\Box\varphi)^a}{\Delta^a; \Sigma^c; (\neg\varphi)^a} \qquad \frac{(\Box\Delta)^a; (\Box\Sigma)^c; (\neg\Box\varphi)^c}{\Delta^a; \Sigma^c; (\neg\varphi)^c}
$$

We deal first with the left hand case. By inductive hypothesis there is a closing tableau for $\Delta^a; (\neg\varphi)^a; (\neg\chi)^c$ and a closing tableau for $\chi^a; \Sigma^c$. The following steps are now valid.

$$
(91) \qquad \frac{(\Box\Delta)^a; (\neg\Box\varphi)^a; (\Box\neg\chi)^c}{\Delta^a; (\neg\varphi)^a; (\neg\chi)^c} \qquad \frac{(\Box\Sigma)^c; (\neg\Box\neg\chi)^a}{\Sigma^c; \chi^a}
$$

The desired interpolant is $\neg\Box\neg\chi$.

Now we look at the right hand case. By inductive hypothesis there is a formula χ in the common variables and a closing tableau for $\Sigma^c; (\neg\varphi)^c; \chi^a$ and one for $\Delta^a; (\neg\chi)^c$. Now look at the following tableaux.

$$
(92) \qquad \frac{(\Box\Delta)^a; (\neg\Box\chi)^c}{\Delta^a; (\neg\chi)^c} \qquad \frac{(\Box\Sigma)^c; (\Box\chi)^a; (\neg\Box\varphi)^c}{\Sigma^c; \chi^a; (\neg\varphi)^c}
$$

So, $\Box\chi$ is the desired interpolant. The other induction cases are dealt with similarly.

For extensions of **K** the tableau methods have proved not so useful. The criterion of [52] is not so easy to apply. Here is another method. Call a function X from sets of formulae to sets of formulae a **local reduction function** from logic L to logic M if the following holds for all Δ and φ:

1. $X(\Delta) \subseteq L$.
2. If Δ is finite, so is $X(\Delta)$.
3. $\mathrm{var}(X(\Delta)) \subseteq \mathrm{var}(\Delta)$.

4. $\Delta \vdash_L \varphi$ iff $\Delta; X(\Delta; \varphi) \vdash_M \varphi$.

A **global reduction function** satisfies the same conditions, with Condition 4 replaced by

(93) $\quad \Delta \Vdash_L \varphi$ iff $\Delta; X(\Delta; \varphi) \Vdash_M \varphi$

The following are global reduction functions to **K**. (For a correct formulation, we assume that the primitive function symbols are $\top$, $\wedge$, $\neg$, and $\Box$. All other symbols are abbreviations. $\mathrm{sf}(\Delta)$ denotes the set of subformulae of formulae from Δ.)

(94) $\quad X_{\mathbf{4}}(\Delta) := \{\Box\chi \to \Box\Box\chi : \Box\chi \in \mathrm{sf}(\Delta)\}$

(95) $\quad X_{\mathbf{T}}(\Delta) := \{\Box\chi \to \chi : \Box\chi \in \mathrm{sf}(\Delta)\}$

(96) $\quad X_{\mathbf{B}}(\Delta) := \{\neg\chi \to \Box\neg\Box\chi : \Box\chi \in \mathrm{sf}(\Delta)\}$

(97) $\quad X_{\mathbf{alt_1}}(\Delta) := \{\neg\Box\chi \to \Box\neg\chi : \Box\chi \in \mathrm{sf}(\Delta)\}$

It is easy to see that reduction functions always exist if $M \subseteq L$. For let $X(\Delta) \subseteq L$. Then from $\Delta; X(\Delta; \varphi) \vdash_M \varphi$ follows $\Delta \vdash_L \varphi$. Conversely, if $\Delta \vdash_L \varphi$, there is a finite proof of φ from Δ. It involves a set $T(\Delta; \varphi)$ of finitely many axioms of L, all of which use only variables from Δ. (To see this, take any proof of φ. If the proof contains a variable q not occurring in φ, replace it uniformly throughout the proof by $\top$. This transforms the proof into a new proof not containing q.) Let $X(\Delta)$ be the union of all these sets $T(\Delta'; \varphi)$ such that $\Delta'; \varphi = \Delta$. This is a reduction function from M to L.

Observe that if X is a local reduction function, then it is also a global reduction function. And if X is a global reduction function, there is a function p from sets of formulae to natural numbers such that $Y(\Delta) := \Box^{\leq p(\Delta)} X(\Delta)$ is a local reduction function. And if Y is a local reduction function, X is a global reduction function.

DEFINITION 100. A reduction function X **splits** if $X(\varphi \to \psi) = X(\varphi) \cup X(\psi)$.

THEOREM 101 (Kracht). *Suppose that there is a splitting global reduction function from L to M. If M has local (global) interpolation, then so does L. If M is locally (globally) Halldén-complete, so is L.*

Proof. Suppose that $\varphi \vdash_L \psi$. Then $\vdash_L \varphi \to \psi$ and so $\Vdash_L \varphi \to \psi$. Hence $X(\varphi \to \psi) \Vdash_M \varphi \to \psi$, and by assumption on X, $X(\varphi); X(\psi) \Vdash_M \varphi \to \psi$. There is a compound modality $\boxplus$ such that $\boxplus X(\varphi); \boxplus X(\psi) \vdash_M \varphi \to \psi$, from which $\boxplus X(\varphi); \varphi \vdash_L (\bigwedge \boxplus X(\psi)) \to \psi$. We have $\mathrm{var}(\boxplus X(\varphi); \varphi) = \mathrm{var}(\varphi)$ and $\mathrm{var}(\boxplus X(\psi) \to \psi) = \mathrm{var}(\psi)$. L has local interpolation, so there is a χ in the joint variables of φ and ψ such that

(98) $\quad \boxplus X(\varphi); \varphi \vdash_M \chi \vdash_M (\bigwedge \boxplus X(\psi)) \to \psi$

From this follows $\varphi \vdash_L \chi \vdash_L \psi$. The proof of global interpolation is similar. Likewise for Halldén-completeness. ❑

The following general result holds (analogous theorems hold for the other functions shown in (94) – (97) with respect to transitive closure (for $X_{\mathbf{4}}$) and reflexive closure (for $X_{\mathbf{T}}$)).

THEOREM 102. *Let L be a complete logic whose class of frames is closed under symmetric closure. Then $L.\mathbf{B}$ is complete for symmetric L-frames. If L has the finite model property, so does $L.\mathbf{B}$. If L has interpolation, so does $L.\mathbf{B}$.*

Proof. Assume that $\Delta \vdash_{L.\mathbf{B}} \varphi$. Put $n := \mathrm{dp}(\Delta;\varphi)$. We show that

$$\Delta; \Box^{\leq n} X_{\mathbf{B}}(\Delta;\varphi) \vdash_L \varphi \tag{99}$$

(In other words, we show that $Y_{\mathbf{B}}(\Delta) := \Box^{\leq \mathrm{dp}(\Delta)} X_{\mathbf{B}}(\Delta)$ is a local reduction function from $L.\mathbf{B}$ to L.) Clearly, if (99) holds, we have $\Delta \vdash_{L.\mathbf{B}} \varphi$. Assume that it fails. Then there is an L-frame $\langle F, \lhd\rangle$ and a model

$$\langle F, \lhd, \beta, x\rangle \vDash \Delta; \Box^{\leq n} X_{\mathbf{B}}(\Delta;\varphi); \neg\varphi \tag{100}$$

Let $\blacktriangleleft$ denote the symmetric closure of $\lhd$. Then $\langle F, \blacktriangleleft\rangle$ is an $L.\mathbf{B}$-frame, by assumption. We claim that for every $\chi \in \mathrm{sf}(\Delta;\varphi)$ and every y accessible in at most $n - \mathrm{dp}(\chi)$ steps from x using $\lhd$ (or in fact $\blacktriangleleft$):

$$\langle F, \blacktriangleleft, \beta, y\rangle \vDash \chi \quad \Leftrightarrow \quad \langle F, \lhd, \beta, y\rangle \vDash \chi \tag{101}$$

The only critical step is $\chi = \Box\chi'$. ($\Rightarrow$) is clear. ($\Leftarrow$). Suppose that we have $\langle F, \blacktriangleleft, \beta, y\rangle \vDash \neg\Box\chi'$. Then there is a z such that $y \blacktriangleleft z$ and $\langle F, \blacktriangleleft, \beta, z\rangle \vDash \neg\chi'$. (1) $y \lhd z$. Then the induction hypothesis yields $\langle F, \lhd, \beta, z\rangle \vDash \neg\chi'$, and the claim follows. (2) $y \ntriangleleft z$. Then $z \lhd y$. Moreover, $\langle F, \lhd, \beta, z\rangle \vDash \neg\chi' \to \Box\neg\Box\chi'$, and so $\langle F, \lhd, \beta, y\rangle \vDash \neg\Box\chi'$.

Finite model property is easy; for interpolation just observe that the global reduction function is splitting. ❑

Notice that it follows that any combination of symmetry, transitivity and reflexivity is covered by these theorems. This can be generalized to polymodal logics. Finally, observe that all the reduction functions split.

THEOREM 103 (Kracht). *Let L be a polymodal logic characterized by any combination of reflexivity, symmetry and transitivity for any of the modal operators. Then L has the finite model property and interpolation.*

This covers among other **K4** and **S4** and fusions thereof. Similarly, passing from a monomodal logic to its minimal tense extension preserves interpolation if the logic is complete. For $\mathbf{alt_1}$ one needs to assume that L is a subframe logic.

Using similar techniques, one can show the following.

THEOREM 104. *Let $L \supseteq \mathbf{K4}$ be a subframe logic with interpolation. Then $L.\mathbf{G}$ and $L.\mathbf{Grz}$ are subframe logics which have interpolation.*

Finally, one can also prove the following observation.

PROPOSITION 105 (Rautenberg). *Let χ be a constant formula. Then if L has local (global) interpolation, so does $L \oplus \chi$.*

5.3 Beth Properties

As is known from predicate logic, interpolation is related to the Beth-property. However, in modal logic the relationship is somewhat more complex.

DEFINITION 106. L has the **local Beth-property** if the following holds. Suppose that $\varphi(p, \vec{q})$ is a formula and

$$\varphi(p,\vec{q}); \varphi(r,\vec{q}) \vdash_L p \leftrightarrow r \tag{102}$$

Then there exists a formula $\chi(\vec{q})$ such that

(103) $\varphi(p, \vec{q}) \vdash_L p \leftrightarrow \chi(\vec{q})$

If (102) is satisfied, $\varphi(p, \vec{q})$ is called a **local implicit definition** of p. If $\chi(\vec{q})$ satisfies (103), it is called the **corresponding explicit definition** of p.

There is a stronger property, the **local projective Beth property**. Here it is required that if

(104) $\varphi(p, \vec{q}, \vec{r}_1); \varphi(p', \vec{q}, \vec{r}_2) \vdash_L p \leftrightarrow p'$

there exists a formula $\chi(\vec{q})$ such that

(105) $\varphi(p, \vec{q}, \vec{r}_1) \vdash_L p \leftrightarrow \chi(\vec{q})$

The global notions are defined similarly. Notice that a local implicit definition is also a global implicit definition. Hence if L has the local Beth-property it also has the global Beth-property.

THEOREM 107 (Maksimova). *A classical modal logic has local interpolation iff it has the local Beth property.*

PROPOSITION 108 (Maksimova). *Let L be a classical modal logic. If L has interpolation then it has the global Beth-property.*

The logic **G.3** has the global Beth-property but fails to have global interpolation. The logic $\mathbf{S4.1.2} \cap \mathbf{S5}$ has the global projective Beth-property but fails to have interpolation ([44]).

5.4 Fixed Point Theorems

A rather different property is shown by logics above **G**. Say that a formula $\psi(\vec{q})$ is a **fixed point** of $\varphi(p, \vec{q})$ for p in L if

(106) $\vdash_L \psi(\vec{q}) \leftrightarrow \varphi(\psi(\vec{q}), \vec{q})$

If a logic has fixed points for all formulae $\varphi(p, \vec{q})$ where p only occurs inside the scope of a $\Box$ (or $\Diamond$) then the logic is said to have the **fixed point property**. It is clear that if $L \subseteq M$ then if $\psi(\vec{q})$ is a fixed point for $\varphi(p, \vec{q})$ for p in L it is one in M, too. The following is known as the **fixed point theorem**.

THEOREM 109 (Sambin, de Jongh). *Suppose that $\varphi(p, \vec{q})$ is a formula in which every occurrence of p is in the scope of a box. Then $\varphi(p, \vec{q})$ has a fixed point for p in* **G**.

Proof. The conditions on $\varphi(p, \vec{q})$ imply that on any finite **G**-frame, the valuation for p is fixed by that for $\vec{q}$. So, $\varphi(p, \vec{q})$ globally implicitly defines p. **G** has local interpolation and so the global Beth property, whence $\psi(\vec{q})$ exists. ❑

It follows that all extensions of **G** have the fixed point property.

THEOREM 110 (Maksimova). *All extensions of* **G** *have the global Beth-property.*

Call a formula φ $\vec{q}$**-boxed** if every occurrence of a variable from $\vec{q}$ is in the scope of some modal operator.

LEMMA 111. *Let L be a logic containing* **G**. *Let q_i, $i < n$, be distinct variables and p a variable not contained in $\vec{q}$. For a set $S \subseteq n$ define χ_S by $\chi_S := \bigwedge_{i \in S} q_i \wedge \bigwedge_{i \in n-S} \neg q_i$. Suppose that $\varphi(p, \vec{q})$ is $\vec{q}$-boxed and that for some S,*

$$\vdash_L \chi_S \to \varphi(p, \vec{q}). \tag{107}$$

Then already $\vdash_L \varphi(p, \vec{q})$.

LEMMA 112. *Let $\varphi(p, \vec{q})$ be a formula. Then there exist $\vec{q}$-boxed formulae $\psi_1(p, \vec{q})$, $\psi_2(p, \vec{q})$, $\chi_1(p, \vec{q})$ and $\chi_2(p, \vec{q})$ such that*

$$\vdash_{\mathbf{G}} \varphi(p, \vec{q}) \leftrightarrow ((p \vee \psi_1(p, \vec{q})) \wedge (\neg p \vee \psi_2(p, \vec{q}))) \tag{108}$$

$$\vdash_{\mathbf{G}} \varphi(p, \vec{q}) \leftrightarrow ((p \wedge \chi_1(p, \vec{q})) \vee (\neg p \wedge \chi_2(p, \vec{q}))) \tag{109}$$

Now suppose that $\varphi(p, \vec{q})$ is a global implicit definition of p in L, and $L \supseteq \mathbf{G}$. Then $\varphi(p, \vec{q}); \varphi(r, \vec{q}) \Vdash_L p \leftrightarrow r$. Using Lemma 112 we get $\vec{q}$-boxed formulae $\chi_1(p, \vec{q})$ and $\chi_2(p, \vec{q})$ such that

$$\vdash_L \varphi(p, \vec{q}) \leftrightarrow ((p \wedge \chi_1(p, \vec{q})) \vee (\neg p \wedge \chi_2(p, \vec{q}))) \tag{110}$$

Write

$$\boxdot\varphi := \varphi \wedge \Box\varphi \tag{111}$$

Since we also have (by transitivity of L) that

$$\vdash_L \boxdot\varphi(p, \vec{q}) \wedge \boxdot\varphi(r, \vec{q}) \to (p \leftrightarrow r) \tag{112}$$

we now get

$$\vdash_L (\Box\varphi(p, \vec{q}) \wedge \Box\varphi(r, \vec{q}) \wedge p \wedge \chi_1(p, \vec{q}) \wedge \neg r \wedge \chi_2(r, \vec{q})) \to (p \to r) \tag{113}$$

This formula has the form $(\mu \wedge p \wedge \neg r) \to (p \to r)$, where μ is $\vec{q}$-boxed. This is equivalent to $\neg\mu \vee \neg p \vee r$, or $(p \wedge \neg r) \to \neg\mu$. By use of Lemma 111 we deduce that $\vdash_L \neg\mu$, that is,

$$\vdash_L \Box\varphi(p, \vec{q}) \wedge \Box\varphi(r, \vec{q}) \to (\chi_1(p, \vec{q}) \to \neg\chi_2(r, \vec{q})) \tag{114}$$

We substitute p for r and obtain

$$\vdash_L \Box\varphi(p, \vec{q}) \to (\chi_1(p, \vec{q}) \to \neg\chi_2(p, \vec{q})) \tag{115}$$

Now from this and (110) it follows after some boolean manipulations

$$\vdash_L \boxdot\varphi(p, \vec{q}) \to \boxdot(p \leftrightarrow \chi_1(p, \vec{q})) \tag{116}$$

By the fixed point theorem for **G** there is a $\psi(\vec{q})$ such that

$$\vdash_{\mathbf{G}} \boxdot(p \leftrightarrow \chi_1(p, \vec{q})) \to (p \leftrightarrow \psi(\vec{q})) \tag{117}$$

So we obtain

$$\vdash_L \boxdot\varphi(p, \vec{q}) \to (p \leftrightarrow \psi(\vec{q})) \tag{118}$$

which is nothing but

(119) $\varphi(p, \vec{q}) \Vdash_L p \leftrightarrow \psi(\vec{q})$

So $\psi(\vec{q})$ is an explicit definition. [1] take a slightly different approach. They show

THEOREM 113 (Areces & Hoogland & de Jongh). *Let L be a transitive logic in which the rule*

(120) $\boxdot p \rightarrow (\Box q \rightarrow q)/ \boxdot p \rightarrow q$

is admissible. Then L has the local Beth property iff it satisfies the fixed point theorem.

Notice that the admissibility of (120) implies that the Löb-rule (121) is admissible.

(121) $\Box p \rightarrow p/p$

For if $\Box\varphi \rightarrow \varphi$ is a theorem, so is $\boxdot\top \rightarrow (\Box\varphi \rightarrow \varphi)$. By (120), $\boxdot\top \rightarrow \varphi$ is a theorem, whence $\varphi \in L$. The following is folklore.

THEOREM 114. *A transitive logic contains* **G** *iff it satisfies the Löb rule.*

Proof. Suppose $L \supseteq \mathbf{G}$ and $\Box\varphi \rightarrow \varphi \in L$. Then $\Box(\Box\varphi \rightarrow \varphi) \in L$, from which $\Box\varphi \in L$. Hence, using (MP$_\rightarrow$) we get $\varphi \in L$. Conversely, assume that the Löb rule is admissible. Put $\chi := \Box(\Box p \rightarrow p)$, $\psi := \Box p$. We need to show that $\chi \rightarrow \psi$ is a theorem of L.

(122)
$$\begin{aligned}
&\vdash_{\mathbf{K4}} \Box(\chi \rightarrow \psi) \rightarrow (\Box\chi \rightarrow \Box\psi)\\
&\vdash_{\mathbf{K4}} \chi \rightarrow (\Box\psi \rightarrow \psi)\\
&\vdash_{\mathbf{K4}} \chi \rightarrow \Box\chi\\
&\vdash_{\mathbf{K4}} \Box(\chi \rightarrow \psi) \rightarrow (\chi \rightarrow \psi)
\end{aligned}$$

Since the Löb rule is admissible in $L \supseteq \mathbf{K4}$, $\chi \rightarrow \psi \in L$. ❑

On the other hand, the same method can be used to show that if $L \supseteq \mathbf{G}$ then (120) is admissible. Suppose namely that $\boxdot\varphi \rightarrow (\Box\chi \rightarrow \chi)$ is a theorem. Then so is

(123) $\Box \boxdot \varphi \rightarrow \Box(\Box\chi \rightarrow \chi)$

From this we get $\Box \boxdot \varphi \rightarrow \Box\chi$ with the **G**-axiom. But $\boxdot\varphi \rightarrow \Box \boxdot \varphi \in \mathbf{G}$, and so $\boxdot\varphi \rightarrow \Box\chi$, which together with the premiss yields $\boxdot\varphi \rightarrow \chi$. Therefore, the coverage of Theorem 113 is not larger than that of Theorems 109 and 110.

5.5 Uniform Interpolation

L has **uniform interpolation** if

① given φ and variables $\vec{q}$ there exists a formula χ such that $\mathrm{var}(\chi) \subseteq \vec{q}$ and for all formulae ψ such that $\varphi \vdash_L \psi$ and $\mathrm{var}(\varphi) \cap \mathrm{var}(\psi) = \vec{q}$ we have $\varphi \vdash_L \chi \vdash_L \psi$ (**uniform preinterpolation**) and

② given ψ and variables $\vec{q}$ there exists a formula χ such that $\mathrm{var}(\chi) \subseteq \vec{q}$ and for all formulae φ such that $\varphi \vdash_L \psi$ and $\mathrm{var}(\varphi) \cap \mathrm{var}(\psi) = \vec{q}$ we have $\varphi \vdash_L \chi \vdash_L \psi$ (**uniform postinterpolation**).

By classical logic, L has uniform preinterpolation iff it has uniform postinterpolation. Notice that uniform interpolation of L can be used to define second order quantification inside the modal language. Let L^q be the extension of L by propositional quantifiers. Now, $(\forall p)\varphi \vdash_L \varphi$ is always valid. Moreover, if $\mathrm{var}(\psi) = \mathrm{var}(\varphi) - \{p\}$ and $\psi \vdash_L \varphi$ then also $\psi \vdash_{L^q} (\forall p)\varphi$. So, $(\forall p)\varphi$ is up to equivalence *the* uniform preinterpolant. If L has uniform interpolation, there is a preinterpolant χ in the variables $\mathrm{var}(\varphi) - \{p\}$. Hence, $(\forall p)\varphi$ is equivalent to χ, and L^q reduces to L in expressivity. This idea has been one of the reasons to study uniform interpolation (see [47]). The logics **K**, **Grz** and **G** have uniform interpolation, **S4** fails to have uniform interpolation (see [57] and [27]). Furthermore, the following is known about fusions, see [62].

THEOREM 115 (Wolter). *If L and L' have uniform interpolation, so does $L \otimes L'$.*

Notice that if $\varphi_1 \vdash_L \psi$ and $\varphi_2 \vdash_L \psi$, and if χ_i are interpolants for φ_i and ψ, then $\chi_1 \vee \chi_2$ is an interpolant for both:

$$(124) \qquad \varphi_1 \vdash_L \chi_1 \vdash_L \chi_1 \vee \chi_2 \vdash_L \psi$$

So if a logic has interpolation and there are up to equivalence only finitely many formulae in n variables then L has uniform interpolation as well ([62]).

THEOREM 116 (Wolter). *Let L have interpolation. If $\mathcal{V}(L)$ is locally finite then L also has uniform interpolation.*

We shall sketch a proof that **K** has uniform interpolation. For example, we show that it has uniform preinterpolation. The proof uses tableau calculi again. By induction on the length of Σ we prove the following: *Let $\vec{q}$ be a set of variables. There is a χ in the variables $\vec{q}$ such that for any Δ such that* $\mathrm{var}(\Delta) \cap \mathrm{var}(\Sigma) = \vec{q}$, *given a closing tableau for $\Delta^a; \Sigma^c$ both $\Delta^a; (\neg\chi)^c$ and $\chi^a; \Sigma^c$ have a closing tableau.* In other words, the interpolant is determined by the ^{c}-set alone (in addition to the set of shared variables). The proof of this fact is actually not hard. We look again at the proof sketched above. Suppose that the tableau closes. It closes in six possible situations. (1) $p^a; (\neg p)^a$, (2) $p^a; (\neg p)^c$, (3) $(\neg p)^a; p^c$, (4) $p^c; (\neg p)^c$, (5) $\bot^a$ and (6) $\bot^c$. In Case (5) $\chi := \bot$ and in Case (6) $\chi := \neg\bot$ satisfy the requirements. Consider the other cases. (A) $p \in \vec{q}$. Then only (2) and (3) can arise. The interpolant is completely determined by knowing Σ. (B) $p \notin \vec{q}$. Then (1) or (4) arise. Again, the interpolant is determined solely by knowing Σ.

We consider briefly the other cases. If (w) has applied, the interpolant χ for the lower sequent is an interpolant for the upper sequent. Clearly, it only depends on the upper ^{c}-set if that was true for the lower ^{c}-set. The same happens with $(\neg E)$ and $(\wedge E)$. Next we look at $(\vee E)$.

$$(125) \qquad \frac{\Delta^a; (\neg(\varphi \wedge \psi))^a; \Sigma^c}{\Delta^a; (\neg\varphi)^a; \Sigma^c \mid \Delta^a; (\neg\psi)^a; \Sigma^c}$$

By inductive hypothesis there is an interpolant χ depending only on Σ^c, not on the ^{a}-set. Therefore we can use the same formula as follows:

$$(126) \qquad \frac{\Delta^a; (\neg(\varphi \wedge \psi))^a; \chi^c}{\Delta^a; (\neg\varphi)^a; \chi^c \mid \Delta^a; (\neg\psi)^a; \chi^c}$$

This tableau closes. Now suppose that the rule application is

$$(127) \qquad \frac{\Delta^a; (\neg(\varphi \wedge \psi))^c; \Sigma^c}{\Delta^a; (\neg\varphi)^c; \Sigma^c \mid \Delta^a; (\neg\psi)^c; \Sigma^c}$$

By inductive hypothesis there are interpolants χ_1 and χ_2 independent of Δ for the left and right hand side. Now we get

(128)
$$\frac{\Delta^a; (\neg(\chi_1 \wedge \chi_2))^c}{\Delta^a; (\neg\chi_1)^c \mid \Delta^a; (\neg\chi_2)^c} \qquad \begin{array}{c|c} \multicolumn{2}{c}{(\chi_1 \wedge \chi_2)^a; (\neg(\varphi \wedge \psi))^c; \Sigma^c} \\ \hline (\chi_1 \wedge \chi_2)^a; (\neg\varphi)^c; \Sigma^c & (\chi_1 \wedge \chi_2)^a; (\neg\psi)^c; \Sigma^c \\ \hline \chi_1^a; \chi_2^a; (\neg\varphi)^c; \Sigma^c & \chi_1^a; \chi_2^a; (\neg\psi)^c; \Sigma^c \\ \hline \chi_1^a; (\neg\varphi)^c; \Sigma^c & \chi_2^a; (\neg\psi)^c; \Sigma^c \end{array}$$

Both tableaux close by assumption. The desired interpolant is $\chi_1 \wedge \chi_2$. So far the interpolant did not depend on what Δ is.

The rule $(\Box E)$ is the last and most complex to consider. Here we face two options: either it was applied to an ^{a}-formula, and then $\neg\Box\neg\chi$ is the new interpolant, or it was applied to a ^{c}-formula, and then the interpolant is $\Box\chi$. Case (1). There is no Δ such that $(\Box E)$ can be applied to an ^{a}-formula. Then the preinterpolant is $\neg\Box\neg\chi$. Case (2). There is no Δ such that $(\Box E)$ can be applied to a ^{c}-formula. Then $\Box\chi$ is the interpolant. Case (3). There is Δ_1 such that $(\Box E)$ can be applied to an ^{a}-formula, and Δ_2 such that $(\Box E)$ can be applied to a ^{c}-formula. Then $\neg\Box\neg\chi \vee \Box\chi$ is the desired interpolant. For by assumption, $(\neg\Box\neg\chi)^a; \Sigma^c$ and $(\Box\chi)^a; \Sigma^c$ both close, and so does therefore $(\neg\Box\neg\chi \vee \Box\chi)^a; \Sigma^c$. And given Δ, either $\Delta^a; (\neg\Box\neg\chi)^c$ closes or $\Delta^a; (\Box\chi)^c$. However, this means that $\Delta^a; (\neg(\neg\Box\neg\chi \vee \Box\chi))^c$ closes.

6 ADMISSIBLE RULES

The study of admissible rules in modal logic has been the topic of the monograph by Vladimir Rybakov, [53], from which most of the results of this section are taken. Studying admissibility can be taken to mean the study of the consequence relations $\vdash_L^m$, where $\vdash_L^m$ is the largest consequence relation whose set of tautologies is L. For in this consequence relation every admissible rule is derived. Thus, we may either speak of characterizing the consequence $\vdash_L^m$ or about the admissible rules of $\vdash_L$, or, for that matter, L itself. We shall prefer the latter. Historically, the first breakthrough was the solution by Rybakov to Problem 40 of the list of 102 problems by Harvey Friedman, [22]. It asked whether admissibility of a rule in **Int** is decidable, which by way of the Gödel translation can be turned into a problem of **Grz**, see Theorem 121. Based on this, Rybakov has extended the results to cover large classes of extensions of **K4**, giving criteria of when admissibility of rules is decidable, and when $\vdash_L^m$ is finitely axiomatisable.

6.1 General Theory

We start with some general considerations. Let $\Delta = \{\delta_i : i < m\}$. A modal algebra satisfies the rule $\langle\Delta, \varphi\rangle$ iff it satisfies the Horn-formula

(129)
$$\bigwedge_{i<m} \delta_i = \top \rightarrow \varphi = \top$$

Admissibility can be characterized as follows. Let L be a logic. $\langle\Delta, \varphi\rangle$ is admissible in L iff for all m

(130)
$$\mathfrak{Fr}_L(n) \vDash \bigwedge_{i<m} \delta_i = \top \rightarrow \varphi = \top$$

where $\mathfrak{Fr}_L(n)$ denotes the freely n-generated L-algebra. For notice that for every valuation h into $\mathfrak{Fr}_L(n)$ there are formulae σ_i, $i \in \omega$, such that $h(p_i) = \sigma_i$. Hence with κ the map induced by the identity and $\sigma : p_i \mapsto \sigma_i$, $h(\varphi) = \kappa(\varphi^\sigma)$. So, if $h(\varphi) = 1$ in $\mathfrak{Fr}_L(n)$ there is a substitution σ for which $\kappa(\varphi^\sigma) = 1$, which means that $\varphi^\sigma \in L$. Equivalently we have

PROPOSITION 117. *Let* $\Delta = \{\delta_i : i < n\}$. $\langle \Delta, \varphi \rangle$ *is admissible in* L *iff* $\mathfrak{Fr}_L(\omega) \vDash \bigwedge_{i<n} \delta_i = \top \to \varphi = \top$.

We shall restrict our attention to extensions of **K4**. The problem whether admissibility of a rule is decidable in intuitionistic logic can be turned into a question of modal logics. Let us note that each rule can be brought into the form $\langle \{\chi_1\}, \chi_2 \rangle$, also written χ_1/χ_2. Now, call a substitution s a **unifier** for χ in L if $\vdash_L s(\chi)$. Then the rule χ_1/χ_2 is admissible in L if every unifier for χ_1 in L is also a unifier for χ_2. Thus admissibility can be checked by inspecting the unifiers of a given formula. In a logic L, say that s is **more general than** s', in symbols $s \leq s'$ if there is a substitution t such that $t(s(p)) \leftrightarrow s'(p) \in L$. Classical logic enjoys the property that if a formula has a unifier, it also has a unique most general unifier and it can effectively be found ([45]). Given that, admissibility can be checked in boolean logic as follows. Determine the most general unifier, say s, for χ_1. Now decide whether $s(\chi_2)$ is a theorem. This fails in intuitionistic logic for the reason that there is no single most general unifier. The strategy can however be generalized. Suppose for any given formula χ we can compute a finite set Π_χ of minimal unifiers, then we can decide admissibility if the logic is decidable. (If L is undecidable, admissibility is a fortiori undecidable.)

[25] gives a proof along these lines that admissibility is decidable in intuitionistic logic. The methods are similar for modal logic. A formula χ is called **projective** if there is a unifier s such that for all $p \in \mathrm{var}(\chi)$:

$$\chi \vdash p \leftrightarrow s(p) \tag{131}$$

It is possible to construct such a unifier. Let S be a subset of $\mathrm{var}(\chi)$. Define θ^S_χ by

$$\theta^S_\chi(p) := \begin{cases} \chi \to p & \text{if } p \in S \\ \chi \wedge p & \text{otherwise} \end{cases} \tag{132}$$

This substitution satisfies (131) but is not necessarily a unifier. Define an enumeration S_i, $i < k$, $k := 2^{|\mathrm{var}(\chi)|}$ of the subsets of $\mathrm{var}(\chi)$ so that if $S_i \subseteq S_j$, then $i \leq j$. Next put

$$\theta_\chi := \theta^{Sk-1}_\chi \circ \theta^{S_{k-2}}_\chi \circ \cdots \circ \theta^{S_0}_\chi \tag{133}$$

THEOREM 118 (Ghilardi). θ_χ *is a unifier for* χ *iff* χ *is projective.*

This serves as a test for projectivity. Define $c(\chi)$ to be the maximum nesting of $\to$ (alternatively, $c(\chi)$ is the $\to$-depth of χ). There are only finitely many formulae χ over a given finite set of variables such that $c(\chi) \leq n$, for any given n. (The other connectives are $\wedge$, $\vee$ and $\neg$. Obviously, this requires showing that from a given set of formulae, there is a bounded number of formulae that can be built using $\neg$, $\vee$ and $\wedge$.) Say that a set U of substitutions is **complete** for χ in L if for every unifier t for χ there is an $s \in U$ such that $s \geq t$. To check the admissibility of a rule χ/φ in L it is enough to be able

to determine whether a formula has a unifier and if so to be able to construct a finite complete set for it. For then it is enough to check the complete set for φ against that for χ.

THEOREM 119 (Ghilardi). *Every unifiable formula has a finite complete set of unifiers in* **Int**.

This set is found as follows. Let

$$S_\chi := \{\psi : \mathrm{var}(\psi) \subseteq \mathrm{var}(\chi), \psi \text{ projective and } c(\psi) \leq c(\chi)\} \tag{134}$$

This set is shown to be finite. Then $\{\theta_\psi : \psi \in S_\chi\}$ is complete. What we really need, though, is a set of substitutions that is a basis, where S is a **basis** iff it is complete and for every $s, t \in S$, if $s \neq t$ then $t \not\leq s$. To get a basis, let Π_χ be any subset of S_χ for which (i) if $\psi_1, \psi_2 \in \Pi_\chi$ and $\psi_1 \vdash \psi_2$ then $\psi_1 = \psi_2$ and (ii) for every $\psi \in S_\chi$ there is a $\psi' \in \Pi_\chi$ such that $\psi \vdash \psi'$. Such a set obviously exists and is easy to construct on the basis of S_χ. The set $\{\theta_\psi : \psi \in \Pi_\chi\}$ is a basis for χ. Now the rule χ/χ' is admissible iff for every $\psi \in \Pi_\chi$: $\psi \vdash \chi'$.

Let us briefly mention some relations with modal logic. Consider the dual of $\mathfrak{Fr}_L(n)$, the weak canonical frame $\mathfrak{Can}_L(n)$. Let $\mathfrak{Ch}_L(n)$ be the subframe of all points of finite depth. This is also called the n-**characterizing frame**, while $\langle \mathfrak{Ch}_L(n), \kappa\rangle$, $\kappa : p_i \mapsto p_i$, is called the n-**characterizing model**.

LEMMA 120. *Assume that $L \supseteq$* **K4** *has the finite model property. Then the rule $\delta_0, \ldots, \delta_{m-1}/\varphi$ is admissible in L iff for all n,*

$$\mathfrak{Ch}_L(n) \vDash \bigwedge_{i<m} \delta_i = \top \to \varphi = \top \tag{135}$$

Recall the Gödel-translation T from intuitionistic logic to modal logic. Let L be a superintuitionistic logic. Put $\sigma(L) :=$ **Grz** $\oplus T(L)$.

THEOREM 121. *The rule $\delta_0, \ldots, \delta_{n-1}/\varphi$ is admissible in L iff the translation*

$$T(\delta_0), \ldots, T(\delta_{n-1})/T(\varphi) \tag{136}$$

is admissible in $\sigma(L)$.

6.2 Frame Characterisation of Admissibility

A logic has **branching below** m if whenever in some frame for L there is a cluster with d immediate successor clusters, then whenever we find d incomparable clusters in $\mathfrak{Ch}_L(n)$, there is a cluster C having these clusters as its immediate successor clusters. The effective m-drop point property is still more cumbersome to define. To understand it, recall the selection procedure of Fine and Zakharyaschev (see [20] and [63]). This procedure extracts a finite model out of a given model $\mathfrak{M}$ on the basis of a set Σ of formulae closed under subformulae. Denote this frame by $X(\mathfrak{M}, \Sigma)$, and by $X_m(\mathfrak{M}, \Sigma)$ the model containing both $X(\mathfrak{M}, \Sigma)$ and the points of depth at most m. (We are assuming that the model is based on finitely many generators.) Crucially, this procedure does not preserve the truth of all formulae, since we are taking not necessarily generated subframes, but it does preserve the truth of all formulae from Σ. For cofinal subframe logics this

shows that they have the finite model property. The m-**drop point property** says the following. Suppose that we have a finite n-generated L-model $\mathfrak{M}$ and that it is large. Then it contains a submodel $\mathfrak{W} \supseteq X_m(\mathfrak{M}, \Sigma)$ which is contractible onto an L-frame of no more than $g(x, y)$ elements, where g is a recursive function and $x = |\Sigma|$, and y the number of points of depth at most m in $\mathfrak{M}$.

THEOREM 122 (Rybakov). *Suppose that L is a logic containing* **K4**. *Suppose further that*

1. *L has fmp,*
2. *L has branching below m for some $m \in \omega$, and*
3. *L has the effective m-drop point property for some $m \in \omega$.*

Let ρ be a rule with k variables. Then ρ is admissible in L iff it is valid in the algebra of all subsets of the Kripke-frame underlying the k-characterizing frame. Furthermore, suppose that there is an algorithm which decides for a finite frame whether it is an L-frame. Then there exists an algorithm deciding whether a given inference rule is admissible for L.

The proof of this theorem uses the selection procedure. It shows that if ρ is refutable in the n-characterizing model then we can construct a model whose size we can estimate a priori and in which ρ is refuted as well. This model also has the so-called **view-realizing** property. Conversely, if such a model exists, ρ is refutable in the n-characterizing model. The proof of the latter statement is the most involved, but it seems that it can be simplified using the technique of homogenisation proposed in [39].

Let I_n be an axiom saying that the frames are of width at most n (that is, have no antichain of length $n + 1$).

COROLLARY 123. *Admissibility of rules is decidable in the modal systems* **K4**, **S4**, **GL**, **Grz**, **S5**, *and in the logics $L \oplus I_n$, where L is any of the aforementioned logics.*

Ghilardi and Sacchetti apply in [26] the method of [25] and develop criteria for extensions L of **K4**. Let us be given a formula χ. There are infinitely many substitutions s such that χ^s is a theorem of L. Say that unification in L is **filtering** if for any two unifiers s_0 and s_1 for a formula χ there is a unifier t such that $t \leq s_0, s_1$. Evidently, if unification is filtering in L then a complete set is either infinite or contains just one member. (If the latter is always the case, L is called **unitary**.)

THEOREM 124 (Ghilardi & Sacchetti). *Let $L \supseteq$* **K4**. *Then unification is filtering iff $L \supseteq \mathbf{K4} \oplus \mathbf{2}^+ := \mathbf{K4} \oplus \neg \boxdot \neg \boxdot p \to \boxdot \neg \boxdot \neg p$.*

The additional axiom is similar $\mathbf{2} = \Diamond\Box p \to \Box\Diamond p$, only that we use $\boxdot$ in place of $\Box$. So, above **S4** this axiom reduces to **2**. The condition also has algebraic analogs. Say that an algebra $\mathfrak{A}$ is **projective** in a variety $\mathcal{V}$ if there is a free algebra $\mathfrak{Fr}_{\mathcal{V}}(X)$ and maps $p : \mathfrak{Fr}_{\mathcal{V}}(X) \to \mathfrak{A}$ and $m : \mathfrak{A} \to \mathfrak{Fr}_{\mathcal{V}}(X)$ such that $p \circ m = 1_{\mathfrak{A}}$. Say that an algebra is **finitely presented** in $\mathcal{V}$ if there is a finite set X and a finite set E of equations such that $\mathfrak{A} \cong \mathfrak{Fr}_{\mathcal{V}}(X)/\Theta(E)$, where $\Theta(E)$ is the smallest congruence containing E.

THEOREM 125 (Ghilardi & Sacchetti). *Unification in L is filtering iff finitely presented projective L-algebras are closed under binary products.*

Let $\mathfrak{F}_i = \langle F_i, \lhd \rangle$ be a family of L-frames. $(\bigoplus_{i \in I} \mathfrak{F}_i)^\circ$ and $(\bigotimes_{i \in I} \mathfrak{F}_i)^\bullet$ are defined like the disjoint union, except that a root world is added, which is irreflexive in the first, and

reflexive in the second case. Finally, $\mathrm{irr}((\bigoplus_{i\in I}\mathfrak{F}_i)^\circ)$ and $\mathrm{irr}((\bigoplus_{i\in I}\mathfrak{F}_i)^\bullet)$ are obtained by identifying all final clusters (assuming that they are isomorphic). Now L has the **2-glueing property** if whenever L has a Kripke-frame containing an irreflexive point (a reflexive point) and $\mathfrak{F}_i$, $i \in I$, are L-frames whose final clusters are isomorphic, then $\mathrm{irr}((\bigoplus_{i\in I}\mathfrak{F}_i)^\circ)$ and $(\mathrm{irr}((\bigoplus_{i\in I}\mathfrak{F}_i)^\circ))$ is an L-frame.

THEOREM 126 (Ghilardi & Sacchetti). *Unification is unitary in $L \supseteq$ **K4** if L contains **K4** $\oplus$ $\mathbf{2}^+$, has the finite model property and has the 2-glueing property.*

In particular, for every L satisfying these conditions the admissibility of inference rules is decidable, which implies that the logics are decidable. For clearly, if admissibility of a rule in L is decidable L must be decidable. But the converse need not hold.

THEOREM 127 (Chagrov). *There is a logic which is decidable, but admissibility of rules is undecidable.*

6.3 Axiomatizing the Admissible rules

There also is a question whether the admissible rules can actually be axiomatized. In the present terms this means axiomatizing $\vdash^m_L$. One speaks of a **basis** for the set of admissible rules. In [53], a series $\mathfrak{E}_n$, $n \in \omega$, of frames is defined.

1. $E^1_n := \{x^0_0\}$, $x^0_0 \lhd_0 x^0_0$.
2. $E^2_n := E^1_1 \cup \{x^1_i : i < 2^n + 2\}$. $x^i_j \lhd_2 x^{i'}_{j'}$ iff $i' = 0$ or $i = i'$ and $j = j'$.
3. Let H be the set of all antichains of $E^i_n - E^{i-1}_n$. $E^{i+1}_n := E^i_n \cup \{x^i_h : h \in H\}$. $\lhd_{i+1}$ satisfies (a) $\lhd_{i+1} \restriction E^i_n = \lhd^i$, (b) $x^i_h \lhd_{i+1} x^j_k$ iff $j = i-1$ and $k \in h$ or there is a x^{i-1}_p such that $p \in h$ and $x^{i-1}_p \lhd_i x^j_k$.

$$\mathfrak{E}_n := \langle \bigcup_i E^i_n, \bigcup_i \lhd_i \rangle \tag{137}$$

Furthermore, the following is established.

THEOREM 128 (Rybakov). *Let $L \supseteq$ **S4** be a logic with the following properties.*

① *For all n: $\mathfrak{F}_n \vDash L$.*

② *L has the finite model property and branching below 1.*

③ *L has the effective m-drop point property.*

Then $\vdash^m_L$ cannot be axiomatized by finitely many rules.

Similar criteria are established for superintuitionistic logics and logics containing **K4**. What is important is the following consequence.

THEOREM 129 (Rybakov). *The logics **S4**, **S4.1** and **S4.2** have no finite basis for the admissible rules.*

[53] also shows that the logics **K4**, **K4.1**, **K4.2**, and **G** have no finite basis for admissible rules.

6.4 Decidability of the Admissibility of a Rule

Lemma 120 can be strengthened. A rule is admissible in L with finite model property iff it is valid in $\mathfrak{Ch}_L(n)$, where n is the number of variables occurring in the rule. We obtain the following.

THEOREM 130 (Rybakov). *Let $L \supseteq$ **K4** be finitely axiomatisable. Suppose that $\mathcal{V}(L)$ is locally finite. Then the admissibility of a given rule in L is decidable.*

6.5 Structural Completeness

For a class $\mathcal{K}$ of algebras, $\mathcal{K}^Q$ denotes the least quasi-variety containing $\mathcal{K}$. The following is a useful criterion.

THEOREM 131 (Rybakov). *A modal logic $L \subseteq$ **K4** is structurally complete iff every subdirectly irreducible $\mathfrak{A} \in \mathcal{V}(L)$ is contained in $(\mathfrak{Fr}_L(\omega))^Q$. This is the case iff $\mathcal{V}(L) = (\mathfrak{Fr}_L(\omega))^Q$.*

If L is a logic that has the finite model property then the free algebra $\mathfrak{Fr}_L(\omega)$ is a subalgebra of the product of the finite subdirectly irreducible L-algebras. Under this condition, a logic L is structurally complete iff every finite subdirectly irreducible L-algebra is embeddable into the algebra $\mathfrak{Fr}_L(\omega)$ (or some $\mathfrak{Fr}_L(n)$, $n \in \omega$). Suppose $\mathfrak{A}$ is a finite, subdirectly irreducible **K4**-algebra. Then $\mathfrak{A}$ has an opremum ω. Now, for each element a of $\mathfrak{A}$ let p_a be a variable and let $r(\mathfrak{A})$ be the following rule:

$$r(\mathfrak{A}) := \frac{\{p_{a*b} \leftrightarrow p_a * p_b : a, b \in A\} \cup \{p_{\circ a} \leftrightarrow \circ p_a : a \in A\} \cup \{p_1\}}{p_\omega} \tag{138}$$

where $*$ runs through all the basic binary connectives and $\circ$ through all the basic unary connectives. This is the **quasi-characteristic** inference rule of $\mathfrak{A}$. Now the following holds:

THEOREM 132 (Citkin). *Let $\mathfrak{A}$ be a finite, subdirectly irreducible* **K4***-algebra. Then for any* **K4***-algebra $\mathfrak{B}$, $r(\mathfrak{A})$ is invalid in $\mathfrak{B}$ iff $\mathfrak{A}$ is isomorphically embeddable into $\mathfrak{B}$.*

This technique is reminiscent of the technique of splittings (see [10] and [11]).

It is not hard to show that no **K4**-algebra with at least two elements is embeddable into $\mathfrak{Fr}_{\mathbf{K4}}(\omega)$. Armed with this result one can show that there are infinitely many admissible rules which are independent from each other. One has to show only that there are infinitely many simple, finite **K4**-algebras. On the other hand, the set of admissible quasi-characteristic rules of **S4** and **Grz** have a finite basis. In the latter case the generalized Mints' rule alone forms a basis:

$$\frac{[(p \to q) \to (q \vee r)] \vee u}{[((p \to q) \to q) \vee ((p \to q) \to r)] \vee u} \tag{139}$$

For **S4** we need in addition to the modal translation of this rule two more, one of which is the quasi-characteristic rule of the two element cluster, which is equivalent to the rule $\Diamond p; \Diamond\neg p/q$.

This can be brought to bear on extensions of **S4.3** in the following way.

LEMMA 133. *Let L be a modal logic containing* **S4.3** *and $\mathfrak{A}$ a finite, subdirectly irreducible L-algebra. Then $\mathfrak{A} \times \mathbf{2}$ is a subalgebra of $\mathfrak{Fr}_L(\omega)$, where $\mathbf{2}$ is the two-element* **S4***-algebra.*

LEMMA 134. *The rule $\Diamond p; \Diamond\neg p/q$ is valid in $\mathfrak{A}$ iff the algebra of the two element cluster is not embeddable into $\mathfrak{A}$.*

Now, any extension L of **S4.3** is finitely axiomatisable and has the finite model property, by results of [7] and [18]. L has the property of branching below 1 and the effective m-drop point property for some m. It follows that the admissibility of inference rules is decidable for L. Second, if we add the rule $\Diamond p; \Diamond\neg p/q$ then the resulting consequence relation axiomatizes the quasi-variety containing all finite L-algebras of the form $\mathfrak{A} \times \mathbf{2}$. Since L is determined by such algebras, we see that this quasi-variety contains $\mathfrak{Fr}_L(\omega)$. Moreover, since the smallest quasi-variety containing $\mathfrak{Fr}_L(\omega)$ must contain these algebras, the two are equal.

THEOREM 135 (Rybakov). *Let $L \supseteq$ **S4.3**. Then $\vdash^m_L$ is axiomatized over $\vdash_L$ by (MN) and $\Diamond p; \Diamond\neg p/q$.*

We derive that **Grz.3** is structurally complete, since the rule $\Diamond p; \Diamond\neg p/q$ is actually derivable in $\Vdash_{\mathbf{Grz.3}}$. It follows that **LC** is also structurally complete, since $\mathbf{Grz.3} = \sigma(\mathbf{LC})$.

Call a logic L **hereditarily structurally complete** if all its extensions are structurally complete. L is **structurally precomplete** if it is not structurally complete, but all its proper extensions are.

THEOREM 136 (Rybakov). *There are exactly 20 structurally precomplete logics containing* **K4***, and they are all tabular of the form* $\mathrm{Th}(\mathfrak{H}_i)$, $i < 20$.

The Kripke-frames $\mathfrak{H}_0 - \mathfrak{H}_{19}$ mentioned in the theorem are known. (They are for example of width and depth at most 3.) We derive the following.

COROLLARY 137. *$L \supseteq$ **K4** is hereditarily structurally complete iff*

(140) $\qquad L \supseteq \mathbf{K4}/\{\mathfrak{H}_i : i < 20\}$

From this, results on **S4** and **Int** can be immediately derived (since the frames are explicitly known). All these logics must be of width 2.

7 FURTHER TOPICS

There are notions of consequence that are not included in this study that we shall mention here only briefly. First, a **multiple conclusion rule** is a pair $\langle\Delta, \Theta\rangle$ of sets of formulae. A multiple conclusion rule is **derived** in $\vdash$ if whenever Δ is made true by a substitution, that substitution makes at least one member of Θ true. It is **admissible** in L if for every substitution such that $\Delta^\sigma \subseteq L$ we have $\theta^\sigma \in L$ for at least one $\theta \in \Theta$. A case in point is the pair $\langle\{p \vee q\}, \{p, q\}\rangle$. This rule is admissible in intuitionistic logic but not in classical logic. Its reflex in modal logic is the rule $\langle\{\Box p \vee \Box q\}, \{\Box p, \Box q\}\rangle$. A modal logic has the **disjunction property** if this rule is admissible. Since the disjunction property does not specify which of the alternatives holds, it is not characterisable in terms of single-conclusion rules. In [33] a modal logic is said to **provide the rule of disjunction** if all of the rules $\langle\{\Box \bigvee_{i<n} p_i\}, \{\Box p_i : i < n\}\rangle$ are admissible, and it is shown that if a logic provides the rule of disjunction then the canonical frame is generated by a single point, which is the set $\{\neg\Box\varphi : \varphi \notin L\}$. More on the disjunction property can be found in [9].

The multiple conclusion rules are more general than ordinary rules, which we may call also **single conclusion rules**.

There is also a **strong rule of disjunction**:

$$(141) \qquad \langle\{\bigvee_{i<n} \Box p_i\}, \{p_i : i < n\}\rangle$$

(see [58]) and the **rule of margins**

$$(142) \qquad \langle\{p \to \Box p\}, \{p, \neg p\}\rangle$$

(see [59]).

Another kind of rule is presented by the **irreflexivity rule**.

$$(143) \qquad \frac{\neg(p \to \Diamond p) \to \varphi}{\varphi} \qquad \text{provided } p \text{ does not occur in } \varphi$$

This rule has been proposed in [23]. It is called irreflexivity rule since in tense logic adding that rule to a logic L gives the logic of the irreflexive frames of L. In ordinary modal logic this does not go through ([56]), unless one adds infinitely many of them (see [24]). See also the chapter on hybrid logics. The difference between this rule and the standard or multiple conclusion rules is the reference to variables, which we know from predicate logic but is quite uncommon elsewhere in propositional logic. The possibility of defining negative properties of frames using rules has been explored in [56].

ACKNOWLEDGEMENTS

My thanks go in particular to Ian Hodkinson for his generous and detailed comments, which led to simplifications in the presentation; and to Yde Venema for many useful hints. All errors and omissions are my own responsibility.

BIBLIOGRAPHY

[1] Carlos Areces, Eva Hoogland, and Dick de Jongh. Interpolation, Definability and Fixed Points. In Michael Zakharyaschev, Krister Segerberg, Maarten de Rijke, and Heinrich Wansing, editors, *Advances in Modal Logic, Volume 2*, pages 35 – 58. CSLI, 2001.
[2] Patrick Blackburn. Nominal tense logic. *Notre Dame Journal of Formal Logic*, 39:56 – 83, 1993.
[3] Wim Blok. The lattice of modal algebras is not strongly atomic. *Algebra Universalis*, 11:285 – 294, 1980.
[4] Wim Blok. On the degree of incompleteness in modal logics and the covering relation in the lattice of modal logics. Technical Report 78 – 07, Department of Mathematics, University of Amsterdam, 1980.
[5] Wim Blok and Don Pigozzi. *Algebraizable Logics.* Number 396 in Memoirs of the American Mathematical Society. 1989.
[6] Wim Blok and Don Pigozzi. On the structure of varieties with equationally definable principal congruences III. *Algebra Universalis*, 32:545 –608, 1994.
[7] Robert A. Bull. That all normal extensions of S4.3 have the finite model property. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 12:341 – 344, 1966.
[8] Alexander Chagrov and Michael Zakharyaschev. On the Independent Axiomatizability of Modal and Intermediate Logics. *Journal of Logic and Computation*, 5:287 – 302, 1995.
[9] Alexander Chagrov and Michael Zakharyaschev. *Modal Logic.* Oxford University Press, Oxford, 1997.
[10] A. I. Citkin. On Admissible Rules for Intuitionistic Propositional Logic. *Math. USSR Sbornik*, 31:279 – 288, 1977.

[11] A. I. Citkin. On Structurally Complete Superintuitionistic Logics. *Math. Doklady*, 19:816 – 819, 1978.
[12] Janusz Czelakowski. Equivalential logics (I). *Studia Logica*, 40:227 – 236, 1981.
[13] Janusz Czelakowski. Equivalential logics (II). *Studia Logica*, 40:335 – 372, 1981.
[14] Janusz Czelakowski. *Protoalgebraic Logics*. Number 10 in Trends in Logic. Kluwer, 2001.
[15] Marcello d'Agostino. Are Tableaux an Improvement on Truth–Tables? Cut–free Proofs and Bivalence. *Journal of Logic, Language and Information*, 1:235 – 252, 1993.
[16] Maarten de Rijke. The modal logic of inequality. *Journal of Symbolic Logic*, 57:566 – 584, 1992.
[17] Maarten de Rijke. *Extending Modal Logic*. PhD thesis, Universiteit van Amsterdam, 1993.
[18] Kit Fine. The logics containing S4.3. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 17:371 – 376, 1971.
[19] Kit Fine. An ascending chain of S4 logics. *Theoria*, 40:110 – 116, 1974.
[20] Kit Fine. Logics containing K4, Part II. *Journal of Symbolic Logic*, 50:619 – 651, 1985.
[21] Josep Maria Font and Ramon Jansana. Leibniz filters and the strong version of a protoalgebraic logic. *Archive for Mathematical Logic*, 40:437 – 465, 2001.
[22] Harvey Friedman. One hundred and two problems in mathematical logic. *Journal of Symbolic Logic*, 40:113 – 129, 1975.
[23] Dov M. Gabbay. An irreflexivity lemma with applications to axiomatizations of conditions on linear frames. In U. Mönnich, editor, *Aspects of philosophical logic*, pages 67 – 89. Reidel, Dordrecht, 1981.
[24] Dov M. Gabbay, Ian Hodkinson, and Mark Reynolds. *Temporal Logic, Vol. 1*. Oxford Logic Guides. Oxford University Press, Oxford, 1994.
[25] Silvio Ghilardi. Unification in intuitionistic logic. *Journal of Symbolic Logic*, 64:859 – 880, 1999.
[26] Silvio Ghilardi and Lorenzo Sacchetti. Filtering unification and most general unifiers in modal logic. *Journal of Symbolic Logic*, 69:879 – 906, 2004.
[27] Silvio Ghilardi and M. Zawadowski. Undefinability of propositional quantifiers in modal system S4. *Studia Logica*, 55:259 – 271, 1995.
[28] George Goguadze, Carla Piazza, and Yde Venema. Simulating polyadic modal logics by monadic ones. *Journal of Symbolic Logic*, 68:419 – 462, 2003.
[29] Valentin Goranko and George Gargov. Modal logic with names. *Journal of Philosophical Logic*, 22:607 – 636, 1993.
[30] George Graetzer and H. Lakser. A note on the implicational class generated by a class of structures. *Canadian Mathematical Bulletin*, 16:603 – 605, 1973.
[31] Sören Halldén. On the semantic non–completeness of certain Lewis calculi. *Journal of Symbolic Logic*, 16:127 – 129, 1951.
[32] R. Harrop. On the existence of finite models and decision procedures for propositional calculi. *Proceedings of the Cambridge Philosophical Society*, 54:1 – 13, 1958.
[33] George E. Hughes and Max J. Cresswell. *A Companion to Modal Logic*. Methuen, London, 1984.
[34] I. L. Humberstone. Inaccessible worlds. *Notre Dame Journal of Formal Logic*, 24:346–352, 1983.
[35] Thomasz Kowalski and Marcus Kracht. Semisimple varieties of modal algebras, *Studia Logica*, 83:1–14, 2006.
[36] Marcus Kracht. An almost general splitting theorem for modal logic. *Studia Logica*, 49:455 – 470, 1990.
[37] Marcus Kracht. Prefinitely Axiomatizable Modal and Intermediate Logics. *Mathematical Logic Quarterly*, 39:301 – 322, 1993.
[38] Marcus Kracht. Modal Logics that Need Very Large Frames. *Notre Dame Journal of Formal Logic*, 40:578 – 587, 1999.
[39] Marcus Kracht. *Tools and Techniques in Modal Logic*. Number 142 in Studies in Logic. Elsevier, Amsterdam, 1999.
[40] Marcus Kracht and Frank Wolter. Normal monomodal logics can simulate all others. *Journal of Symbolic Logic*, 64:99 – 138, 1999.
[41] David C. Makinson. Some embedding theorems for modal logic. *Notre Dame Journal of Formal Logic*, 12:252 – 254, 1971.
[42] Larisa L. Maksimova. Craig's theorem in superintuitionistic logics and amalgamable varieties of pseudo–boolean algebras. *Algebra i Logika*, 16:643 – 681, 1977.
[43] Larisa L. Maksimova. Interpolation theorems in modal logic and and amalgamable varieties of topoboolean algebras. *Algebra i Logika*, 18:556 – 586, 1978.
[44] Larissa Maksimova. Projective Beth's Properties in Infinite Slice Extensions of the Modal Logic K4. In Frank Wolter, Heinrich Wansing, Maarten de Rijke, and Michael Zakharyaschev, editors, *Advances in Modal Logic, Volume 3*, pages 349 – 363. World Scientific, 2002.
[45] Ursula Martin and Tobias Nipkow. Boolean unification — the story told so far. *Journal of Symbolic Computation*, 7:275 – 293, 1989.
[46] Ralph McKenzie. Equational bases and nonmodular lattice varieties. *Transactions of the American Mathematical Society*, 174:1 – 43, 1972.

[47] A. Pitts. On an interpretation of second order quantification in first order intuitionistic logic. *Journal of Symbolic Logic*, 57:33 – 52, 1992.

[48] T. Prucnal and A. Wroński. An algebraic characterization of the notion of structural completeness. *Bulletin of the Section of Logic of the Polish Academy of Sciences*, 3:20 – 33, 1974.

[49] Wolfgang Rautenberg. Der Verband der normalen verzweigten Modallogiken. *Mathematische Zeitschrift*, 156:123 – 140, 1977.

[50] Wolfgang Rautenberg. *Klassische und nichtklassische Aussagenlogik.* Vieweg, Braunschweig/Wiesbaden, 1979.

[51] Wolfgang Rautenberg. Splitting lattices of logics. *Archiv für Mathematische Logik*, 20:155 – 159, 1980.

[52] Wolfgang Rautenberg. Modal tableau calculi and interpolation. *Journal of Philosophical Logic*, 12:403 – 423, 1983.

[53] Vladimir V. Rybakov. *Admissibility of Inference Rules.* Number 136 in Studies in Logic. Elsevier, Amsterdam, 1997.

[54] George F. Schumm. Why does Halldén–completeness matter? *Theoria*, 59:192 – 206, 1993.

[55] Stanisław Surma. The growth of logic out of the foundations of modern mathematics. In Evandro Agazzi, editor, *Modern Logic*, pages 15 – 33. Reidel, Dordrecht, 1981.

[56] Yde Venema. Derivation rules as anti–axioms in modal logic. *Journal of Symbolic Logic*, 58:139 – 164, 1993.

[57] Albert Visser. Uniform interpolation and layered bisimulation. In Peter Hajek, editor, *Gödel 96*, pages 139 – 164. Springer, 1996.

[58] Timothy Williamson. An alternate rule of disjunction in modal logic. *Notre Dame Journal of Formal Logic*, 33, 1992.

[59] Timothy Williamson. Some Admissible Rules in Nonnormal Modal Logics. *Notre Dame Journal of Formal Logic*, 34:378 – 400, 1993.

[60] Ryszard Wójcicki. *Theory of Logical Calculi.* Kluwer, Dordrecht, 1988.

[61] Frank Wolter. *Lattices of Modal Logics.* PhD thesis, FU Berlin, 1993.

[62] Frank Wolter. Independent fusions of modal logics revisited. In Marcus Kracht, Maarten de Rijke, Heinrich Wansing, and Michael Zakharyaschev, editors, *Advances in Modal Logic*, pages 361 – 379, Stanford, 1997. CSLI.

[63] Michael Zakharyaschev. Canonical formulas for K4, Part I: Basic results. *Journal of Symbolic Logic*, 57:1377 – 1402, 1992.

[64] Michael Zakharyaschev. Canonical formulas for K4, Part II: Cofinal Subframe Logics. *Journal of Symbolic Logic*, 61:421 – 449, 1996.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

9

FIRST-ORDER MODAL LOGIC

Torben Braüner and Silvio Ghilardi

1	Introduction to Part I	550
2	The Basic Logics	551
2.1	Syntax of the basic logics	552
2.2	Constant domain semantics	552
2.3	An axiom system for constant domains	553
2.4	Increasing domain semantics	556
2.5	An axiom system for increasing domains	557
2.6	Varying domain semantics	557
2.7	Axiom systems for varying domains	558
2.8	Existence and quantification	559
2.9	The Barcan and Converse Barcan formulas	561
2.10	Translation into two-sorted first-order logic	562
3	First-Order Intensional Logic	564
3.1	Syntax and semantics of first-order intensional logic	565
3.2	Equality and intensions	568
3.3	Predicate abstraction	569
3.4	Translation of first-order intensional logic	570
4	First-Order Hybrid Logic	572
4.1	Syntax and semantics of first-order hybrid logic	573
4.2	Axioms for first-order hybrid logic	575
4.3	Translation of first-order hybrid logic	576
5	Other survey articles and books	578
6	Introduction to Part II	579
7	Decision Problems	580
8	Completeness, Incompleteness and Non-Axiomatizability	586
9	Towards more Powerful Semantics	591
9.1	Typed Languages and Systems	591
9.2	Cartesian and Lex Categories	594
9.3	Hyperdoctrines	596
9.4	Metaframes	599
10	Mathematical Models for Modalities	600
10.1	Geometric Morphisms and Modal Logic	601
10.2	The Need for the Continuity Axiom	605
11	Counterpart Semantics	606
11.1	Counterpart Models	607
11.2	Soundness and Completeness	610
11.3	Conclusions	613

PREFACE

This chapter is divided into two parts. The first (which consists of Sections 1–5) was written by Torben Braüner, and the second (which consists of Sections 6–11) was written by Silvio Ghilardi.

In the first part we give an introduction to first-order modal logic. We discuss a number of logics that make use of constant domain, increasing domain, and varying domain semantics, and also present a first-order intensional logic and a first-order version of hybrid logic. One criterion for selecting these logics has been the availability of sound and complete proof procedures for them, typically axiom systems and/or tableau systems. We compare the first-order modal logics discussed here to fragments of sorted first-order logic via appropriate versions of the standard translation.

In the second part of the chapter, we review both positive and negative results concerning fragment decidability, Kripke completeness and axiomatizability. Modal hyperdoctrines are then introduced, as a unifying tool for analyzing the alternative semantics proposed in the literature. These alternative semantics range from specific semantics for non-classical logics (like metaframes), to interpretations in well-established mathematical frameworks (like topological spaces and toposes). Finally, the strict relationship between topological semantics and D. Lewis's counterpart semantics is investigated in detail and an axiomatization is presented.

1 INTRODUCTION TO PART I

In Part I of this chapter we give an introduction to first-order modal logic. First-order modal logic is a big area with a great number of different logics. This has forced us to make a number of choices. The first choice we made was to concentrate on presenting an appropriate selection of logics rather than trying to be encyclopedic. This has allowed us to give reasonably detailed treatments of each of the selected logics. How did we select the logics in question? We wanted to present logics involving constant domains, increasing domains, and varying domains, and moreover, we wanted to present a first-order intensional logic as well as a first-order version of hybrid logic.

Given these overall requirements, one criteria for selecting particular logics for presentation has been the availability of sound and complete proof procedures, typically axiom systems and/or tableau systems. We have compared the first-order modal logics under consideration to fragments of sorted first-order logic via appropriate versions of the standard translation. The possibility of doing so in a straightforward and simple way has been another criteria for selecting a particular logic for presentation. In fact, we take such a simplicity as a sign of mathematical naturality. We have not included constant symbols in the presented logics, the reason being that constants (having a given sort of semantic values) from a mathematical, model-theoretic point of view are just variables (ranging over the same semantic values) that are not quantified over. In the interest of simplicity, we have not included function symbols either. Counterpart semantics, which can be considered an alternative to first-order intensional logics, is treated in Part II, Section 11 of this chapter.

So, which logics have been chosen? In Section 2 we shall present three first-order modal logics which we call the basic logics. In the basic logics, variables designate individual objects. The three basic logics have respectively constant domains, increasing

domains, and varying domains. In Section 3 we present first-order intensional logic. Beside objectual variables, this logic involves intension variables, that is, variables that designate functions from worlds to individual objects. Thus, there are two different types of semantic values, namely objects and functions from worlds to objects. First-order intensional logic allows quantification over both types of semantic values and predicates are allowed to take both types of semantic values as arguments. The logic includes so-called predicate abstraction which allows the functions from worlds to objects to be applied to arguments, that is, worlds, whereby objects are obtained. In Section 4 we present first-order hybrid logic which is obtained by adding to basic first-order modal logic further expressive power in the form of a new sort of propositional symbol called a nominal, and moreover, by adding so-called satisfaction operators and binders.

2 THE BASIC LOGICS

The syntax of the basic logics considered in this section is simply the syntax of ordinary (non-modal) first-order logic with equality, extended with a modal operator. Variables in the basic logics designate *rigidly*, that is, a variable designate the same object in all worlds. On the other hand, a predicate might have different extensions in different worlds, thus, predicates are relativised to worlds. This allows us to formalise natural language sentences involving predicates like for example "is a citizen of the United States". The fact that this predicate has different extensions in different worlds follows for example from the observation that Arnold Schwarzenegger is a citizen of the United States, but he might not have been so, for example if he had not emigrated to the United States. Predicates with different extensions in different worlds should be compared to predicates which naturally are taken to have the same extension in all worlds, one example being "is greater than five" since the extension of this predicate in any world naturally is taken to be the set of numbers greater than five. Predicates of the latter kind can be formalised in ordinary first-order logic.

One choice to make is whether quantified domains might be different in different worlds, that is, whether quantifier domains are relativised to worlds, and if they are relativised to worlds, whether any restrictions are imposed on this relativisation. In this section we present three different first-order modal logics corresponding to three different choices concerning relativisation, they have respectively constant domains (no relativisation), increasing domains (relativisation with the restriction that the domain of a world is included in the domain of any accessible world), and varying domains (unrestricted relativisation). Note that in all three cases an object might be a member of the quantified domains of more than one world, in fact, in the constant domain case all worlds have the same quantifier domain. In Subsection 2.8 we compare constant and varying domains using so-called *existence predicates*. The choice of relativisation of quantifier domains is related to the famous *Barcan* and *Converse Barcan* formulas, which we shall return to a number of times, in particular in Subsection 2.9.

Sound and complete proof procedures are available. In this section we give axiom systems for the constant, increasing, and varying domain basic logics. A tableau system for a version of the constant domain logic without equality can be found in Chapter 2 in the present handbook, and tableau systems for the constant and varying domain logics, including equality, can be found in the book [41]. For more on the model theory of first-order modal logic, see [30], [31], and [33].

2.1 Syntax of the basic logics

The three basic first-order modal logics have the same syntax, that is, the same formulas. The syntax is obtained by adding a modal operator $\Box$ to the syntax of ordinary first-order logic with equality. It is assumed that a countably infinite set of first-order variables is given. The metavariables x, y, z, ... range over variables. We do not consider function symbols or constants, so all terms are variables. It is also assumed that a set of predicate symbols is given. The metavariables P, Q, R, ... range over predicate symbols. Each predicate symbol comes together with a specification of its arity. Of course, 0-place predicate symbols correspond to propositional symbols. Formulas are defined by the grammar

$$S \ ::= \quad P(x_1, \ldots, x_n) \mid x = y \mid S \wedge S \mid \neg S \mid \Box S \mid \forall x S$$

where P is an n-place predicate symbol, x_1, ..., x_n as well as x and y are variables. Thus, formulas are built in the usual way using the connectives of ordinary first-order logic together with the modal operator. We allow parentheses to be inserted in formulas where needed and we shall assume the usual precedence rules for our logical connectives. In what follows, the metavariables ϕ, ψ, θ, ... range over formulas. Other propositional and first-order connectives such as $\top$, $\vee$, $\bot$, $\rightarrow$, $\leftrightarrow$, and $\exists$ are defined in the usual way. Also the modal operator $\Diamond$ is defined as usual. The notions of free and bound occurrences of variables are defined in the obvious way. Moreover, if $\overline{x}$ is a list of distinct variables and $\overline{y}$ is a list of variables of the same length as $\overline{x}$, then $\psi[\overline{y}/\overline{x}]$ is the formula ψ where the variables $\overline{y}$ have been simultaneously substituted for all free occurrences of the variables $\overline{x}$. It is assumed that no variable x_i in $\overline{x}$ occurs free in ψ within the scope of $\forall y_i$.

2.2 Constant domain semantics

We now define constant domain models and constant domain skeletons. Skeletons are first-order versions of the usual frames for propositional modal logics. We shall in many cases adopt the terminology of the books [11] and [41].

DEFINITION 1. A *constant domain model* is a tuple $\langle W, R, D, \{V_w\}_{w \in W}\rangle$ where

1. W is a non-empty set;
2. R is a binary relation on W;
3. D is a non-empty set; and
4. for each w, V_w is a function that to each n-place predicate symbol assigns a subset of D^n.

The tuple $\langle W, R\rangle$ is called a *frame* and the model is said to be *based* on this frame. The tuple $\langle W, R, D\rangle$ is called a *constant domain skeleton* and the model is said to be *based* on this constant domain skeleton.

As usual, the elements of the set W are called *worlds*, the relation R is called an *accessibility relation*, the set D is called the *domain of quantification*, and the function V_w is called the *valuation* at the world w.

DEFINITION 2. Given a constant domain model $\mathfrak{M} = \langle W, R, D, \{V_w\}_{w \in W}\rangle$, an *assignment* is a function that to each variable assigns an element of D. Given assignments g'

and g, $g' \overset{x}{\sim} g$ means that g' agrees with g on all variables save possibly x. The relation $\mathfrak{M}, g, w \models \phi$ is defined by induction, where w is a world, g is an assignment, and ϕ is a formula of first-order modal logic.

$$\begin{array}{rcl}
\mathfrak{M}, g, w \models P(x_1, \ldots, x_n) & \text{iff} & (g(x_1), \ldots, g(x_n)) \in V_w(P) \\
\mathfrak{M}, g, w \models x = y & \text{iff} & g(x) = g(y) \\
\mathfrak{M}, g, w \models \phi \wedge \psi & \text{iff} & \mathfrak{M}, g, w \models \phi \text{ and } \mathfrak{M}, g, w \models \psi \\
\mathfrak{M}, g, w \models \neg\phi & \text{iff} & \text{not } \mathfrak{M}, g, w \models \phi \\
\mathfrak{M}, g, w \models \Box\phi & \text{iff} & \text{for any } v \in W \text{ where } wRv,\ \mathfrak{M}, g, v \models \phi \\
\mathfrak{M}, g, w \models \forall x\phi & \text{iff} & \text{for any } g' \overset{x}{\sim} g,\ \mathfrak{M}, g', w \models \phi
\end{array}$$

A formula ϕ is said to be *true* at the world w if $\mathfrak{M}, g, w \models \phi$; otherwise it is said to be *false* at w. By convention $\mathfrak{M}, g \models \phi$ means $\mathfrak{M}, g, w \models \phi$ for every world w and $\mathfrak{M} \models \phi$ means $\mathfrak{M}, g \models \phi$ for every assignment g. A formula ϕ is *valid* in a frame (skeleton) if and only if $\mathfrak{M} \models \phi$ for any model $\mathfrak{M}$ that is based on the frame (skeleton) in question. A formula ϕ is *valid* in a class of frames (skeletons) $\mathfrak{F}$ if and only if ϕ is valid in any frame (skeleton) in $\mathfrak{F}$. A formula ϕ is *valid* if and only if ϕ is valid in the class of all frames (or equivalently, in the class of all skeletons).

Let us take a look at a natural language sentence that can be formalised using the machinery introduced so far. Consider the sentence

> Arnold Schwarzenegger is a citizen of the United States.

About Arnold Schwarzenegger, it says that he is a member of the set of persons who happen to be citizens of the United States. If the variable x stands for "Arnold Schwarzenegger" and the 1-place predicate symbol P stands for the predicate "is a citizen of the United States", then the formula $P(x)$ formalises the statement. Formally, $P(x)$ is true at a world w if and only if the designation of x belongs to the extension of the predicate symbol P at w. The relativisation of P to worlds formalises that the predicate "is a citizen of the United States" has different extensions in different worlds.

In ordinary (non-modal) first-order logic, the equality predicate is a designated primitive 2-place predicate symbol which is given a fixed interpretation, namely the identity relation on the domain of quantification. Note that the same pattern is followed in the case of first-order modal logic.

2.3 An axiom system for constant domains

In this subsection we shall give a Hilbert-style axiom system for the constant domain basic first-order modal logic. The axiom system is obtained as an extension of a Hilbert-style axiom system for the propositional modal logic **K** or another propositional modal logic. So we are actually giving a family of constant domain axiom systems, depending on a choice of the underlying propositional axiom system. In the definition of an axiom system we shall make use of *formula-schemas*. Informally, a formula-schema is like an ordinary formula except that it has metavariables for formulas instead of ordinary atomic formulas. Formally, a grammar for formula-schemas can be obtained from the grammar for formulas given in Subsection 2.1 by replacing the clauses for atomic formulas by a clause for metavariables. A *substitution-instance* of a formula-schema is a formula obtained by uniformly replacing all metavariables by ordinary formulas. Using this terminology, we

are able to talk formally about formulas having a specific common form (which is the form of an axiom schema).

We first give an axiom system for the propositional modal logic **K**. The axioms of the system are all the substitution-instances of tautologies of propositional logic (where propositional symbols are considered as metavariables) together with all substitution-instances of the following axiom schema

$$(K) \qquad \Box(\phi \to \psi) \to (\Box\phi \to \Box\psi)$$

The rules of the system are the following

$$\frac{\phi \to \psi \qquad \phi}{\psi}\,(\textit{Modus Ponens}) \qquad\qquad \frac{\phi}{\Box\phi}\,(\textit{Necessitation})$$

If there exists a derivation of a formula in the axiom system given above, then we say that the formula in question is *derivable*. This axiom system is sound and complete with respect to the standard possible-worlds semantics for propositional modal logic, see Chapter 2 in the present handbook. We here make use of the usual definition of soundness and completeness:

DEFINITION 3. An axiom system is *sound* with respect to a semantics if every derivable formula is valid and the axiom system is *complete* with respect to the semantics if every valid formula is derivable.

Further axiom schemas can be added to the axiom system for **K**, for example

$$\begin{array}{ll} (T) & \Box\phi \to \phi \\ (4) & \Box\phi \to \Box\Box\phi \\ (B) & \phi \to \Box\Diamond\phi \end{array}$$

whereby axiom systems for other propositional modal logics are obtained. If for example the axiom schema (T) is added, then an axiom system for the propositional modal logic **T** is obtained. Similarly, if (T) as well as (4) are added, then an axiom system for **S4** is obtained, and if all three axiom schemas above are added, the an axiom system for **S5** is obtained. It is straightforward to show that the axiom schemas above corresponds to the first-order conditions reflexivity, transitivity, and symmetry on the accessibility relation of a frame in the sense that a frame validates all substitution-instances of an axiom schema if and only if the accessibility relation of the frame satisfies the first-order condition corresponding to the axiom schema in question (for example, a frame validates all substitution-instances of the axiom schema (T) if and only if the accessibility relation of the frame is reflexive). Moreover, any axiom system obtained by adding some or all of the axiom schemas above can be proven to be sound and complete with respect to the semantics obtained by relativising validity to the class of frames where the accessibility relations satisfy the first-order conditions corresponding to the added axiom schemas, see Chapter 2 in the present handbook.

We are now ready to give the axiom system for the constant domain basic first-order modal logic. We choose **K** as the underlying propositional modal logic. The system is

obtained by extending the axiom system for propositional **K** with the axiom schemas

(*Reflexivity*)	$x = x$
(*Substitutivity*)	$(x = y \wedge \phi[x/z]) \rightarrow \phi[y/z]$
(*Necessary Distinctness*)	$x \neq y \rightarrow \Box(x \neq y)$
(*Barcan*)	$\forall x \Box \phi \rightarrow \Box \forall x \phi$
($\forall$ *Elimination*)	$\forall x \phi \rightarrow \phi[y/x]$

and the rule

$$\frac{\psi \rightarrow \phi[y/x]}{\psi \rightarrow \forall x \phi} \ (\forall \ \textit{Introduction})$$

where the rule is equipped with the side-condition that the variable y does not occur free in ψ or in $\forall x \phi$. This axiom system is sound and complete with respect to the constant domains semantics given in the previous subsection. A completeness proof can be found in the book [68]. See also the chapter [49] in *Handbook of Philosophical Logic*. This completeness result is essentially due to Saul Kripke, cf. the paper [75]. Soundness and completeness also holds for a number of other systems where further axiom schemas have been added and where the appropriate frame classes are considered, this is for example the case with the modal logics **T**, **S4**, and **S5** mentioned above, cf. the book [68].

The axiom (*Barcan*) is often simply called the Barcan formula. We shall return to it in Subsection 2.9. Note that ϕ in the axiom (*Substitutivity*) can be any formula of first-order modal logic, also a formula that involves modal operators. Thus, we here allow substitution of equals for equals in modal contexts, so for example the formula $x = y \rightarrow \Box(x = y)$ is derivable. This is justified by the fact that variables here designate rigidly. Thus, the variables x and y designating the same object in a world w imply that x and y designate the same object in any world, in particular any world accessible from w. This gives rise to a philosophical discussion which we shall return to in Subsection 3.2.

Note that the axiom ($\forall$ *Elimination*) together with the rule ($\forall$ *Introduction*) is a standard axiomatisation of quantifiers in ordinary (non-modal) first-order logic, and similarly, the axioms (*Reflexivity*) and (*Substitutivity*) together is a standard axiomatisation of equality in ordinary first-order logic, so the constant domain axiom system given above can be seen as obtained from the axiom system for the propositional modal logic **K** by adding standard non-modal axiomatic machinery for quantifiers and equality together with the axioms (*Necessary Distinctness*) and (*Barcan*). This motivates the following definition:

DEFINITION 4. The constant domain axiom system given above will (where such notation is needed) be denoted $\mathbf{QK}_{=}$+*ND*+*BF*. We will (again, where such notation is needed) use the same notation for the set of formulas derivable in the axiom system. The notation is adjusted as appropriate if another underlying propositional modal logic is chosen or if for example the axiom (*Barcan*) is omitted.

In Subsection 2.5 we shall consider the system $\mathbf{QK}_{=}$+*ND* without the axiom (*Barcan*). From Section 9 on in Part II of this chapter we shall treat logics along the lines of $\mathbf{QK}_{=}$, that is, logics with equality but without the axiom (*Necessary Distinctness*). In particular, a semantics for $\mathbf{QK}_{=}$ is given in the first example of Subsection 10.1.

Above we specified an axiom system by specifying the axiom schemas and the rules of the system. Given an axiom system, we could then talk about the set of formulas derivable in the axiom system. Sometimes we want to be more general in the sense that

we want talk about an arbitrary set of formulas containing all substitution instances of some axiom schemas and closed under some rules. To this end we need one more rule

$$\frac{\phi}{\phi[\psi/P(\overline{x})]}\ (\textit{Uniform Substitution})$$

where $\phi[\psi/P(\overline{x})]$ is the formula obtained by replacing every occurrence of $P(\overline{y})$ in ϕ by $\psi[\overline{y}/\overline{x}]$. It is assumed that $P(\overline{y})$ does not occur in ϕ within the scope of any quantifier $\forall z$ where z is a free variable of ψ which is not on the list $\overline{x}$. We shall call $\phi[\psi/P(\overline{x})]$ a *substitution-instance* of ϕ. It might be instructive to consider an example of a substitution-instance: If ϕ is the formula $\forall y(P(y) \wedge Q(y))$ and ψ is the formula $R(z, x)$ where z is distinct from x and y, then $\phi[\psi/P(x)]$ is the formula $\forall y(R(z, y) \wedge Q(y))$. Thus, $\forall y(R(z, y) \wedge Q(y))$ is a substitution-instance of $\forall y(P(y) \wedge Q(y))$. Note that the definition of a substitution-instance of a formula is more complicated than the definition of a substitution-instance of a formula-schema, the reason being that an atomic formula $P(\overline{x})$ has an inner structure whereas the metavariables in a formula-schema work as place-holders, that is, they are just indexes of places. Now, if L is a set of formulas which contains all substitution-instances of the axiom schemas of the axiom system $\mathbf{QK}_=$ and which is closed under the rules of $\mathbf{QK}_=$ together with the rule (*Uniform Substitution*), then the set L is called a *first-order modal system*. With this definition it is straightforward that $\mathbf{QK}_=$ is the least first-order modal system with respect to inclusion. We shall return to this in Section 8 in Part II of this chapter.

2.4 Increasing domain semantics

We now define increasing domain models and increasing domain skeletons. We do not repeat conventions and definitions that are the same as in the constant domain case.

DEFINITION 5. An *increasing domain model* is a tuple $\langle W, R, D, \{\delta_w\}_{w\in W}, \{V_w\}_{w\in W}\rangle$ where the tuple $\langle W, R, D, \{V_w\}_{w\in W}\rangle$ is a constant domain model as defined in Definition 1 and where for each w, δ_w is a subset of D such that $\delta_w \subseteq \delta_v$ whenever wRv. The tuple $\langle W, R, D, \{\delta_w\}_{w\in W}\rangle$ is called an *increasing domain skeleton* and the model is said to be *based* on this increasing domain skeleton.

The set δ_w is called the *domain of quantification* at the world w. Note that a constant domain model (skeleton) can be considered as an increasing domain model (skeleton) by letting $\delta_w = D$ for any element w of W (see Definition 1).

DEFINITION 6. Given an increasing domain model $\mathfrak{M} = \langle W, R, D, \{\delta_w\}_{w\in W}, \{V_w\}_{w\in W}\rangle$, the relation $\mathfrak{M}, g, w \models \phi$ is defined in the same way as in the constant domain case, that is, Definition 2, except that the clause for the quantifier is replaced by

$$\mathfrak{M}, g, w \models \forall x\phi \quad \text{iff} \quad \text{for any } g' \stackrel{x}{\sim} g \text{ where } g'(x) \in \delta_w,\ \mathfrak{M}, g', w \models \phi$$

The definition of validity is the same as in the constant domain case except that $\mathfrak{M} \models \phi$ now means $\mathfrak{M}, g, w \models \phi$ for every world w and every assignment g such that $g(x)$ is an element of δ_w for every variable x.

Of course, if $\delta_w = D$ for any element w of W, then the clause above for the increasing domain quantifier is equivalent to the clause for the constant domain quantifier (compare Definition 2). Note that in the increasing domain semantics, the only assignments considered are assignments where every variable is assigned an existent, that is, an element

of the quantifier domain. Also, note that if an assignment at some world has the property that it assigns existents to all variables, then the assignment also has that property at any accessible world, the reason being that the domain increases. This means that worlds with empty quantifier domains simply are ignored. In a similar way the semantics also ignores elements of D that are not elements of the set $\cup_{w\in W}\delta_w$.

2.5 An axiom system for increasing domains

In this subsection we shall consider a Hilbert-style axiom system for the increasing domain basic first-order modal logic. The system is obtained from the axiom system for constant domains of Subsection 2.3 simply by omitting the Barcan formula $\forall x\Box\phi \rightarrow \Box\forall x\phi$. This axiom system is sound and complete with respect to the increasing domain semantics given in the previous subsection, cf. the book [68]. Note that this axiom system can be seen as obtained from the axiom system for the propositional modal logic **K** by adding standard non-modal axiomatic machinery for quantifiers and equality together with the axiom (*Necessary Distinctness*). So in the terminology of Subsection 2.3, it is the system $\mathbf{QK}_{=}+ND$.

Now, the Barcan formula, which was taken as an axiom in the constant domain system, expresses interaction between first-order quantifiers and modal operators since it says that a quantifier and a modal operator can be permuted in one way. However, the Barcan formula is not taken as an axiom in the system considered here, and there are no other axioms or rules in the system that explicitly say that quantifiers and modal operators interact. It is therefore a surprise that the so-called Converse Barcan formula $\Box\forall x\phi \rightarrow \forall x\Box\phi$ is derivable in the system. In fact, the Converse Barcan formula has the very simple derivation

$$\cfrac{\cfrac{\dfrac{\forall x\phi \rightarrow \phi}{\Box(\forall x\phi \rightarrow \phi)} \qquad \Box(\forall x\phi \rightarrow \phi) \rightarrow (\Box\forall x\phi \rightarrow \Box\phi)}{\Box\forall x\phi \rightarrow \Box\phi}}{\Box\forall x\phi \rightarrow \forall x\Box\phi}$$

in the system. We shall return to the Barcan and Converse Barcan formulas in Subsection 2.9.

2.6 Varying domain semantics

We now define varying domain models and varying domain skeletons. Again, we do not repeat conventions and definitions that are the same as in the earlier cases.

DEFINITION 7. A *varying domain model* is a tuple $\langle W, R, D, \{\delta_w\}_{w\in W}, \{V_w\}_{w\in W}\rangle$ where the tuple $\langle W, R, D, \{V_w\}_{w\in W}\rangle$ is a constant domain model as defined in Definition 1 and where for each w, δ_w is a subset of D. The tuple $\langle W, R, D, \{\delta_w\}_{w\in W}\rangle$ is called a *varying domain skeleton* and the model is said to be *based* on this varying domain skeleton. A varying domain model (skeleton) has *increasing domains* if and only if $\delta_w \subseteq \delta_v$ whenever wRv, and similarly, it has *decreasing domains* if and only if $\delta_w \supseteq \delta_v$ whenever wRv.

Note that an increasing domain model (skeleton) is a varying domain model (skeleton) with increasing domains and vice versa (see Definition 5). A number of choices in the

definition of a varying domain model should be noted: We do not require that a predicate is false of non-existents, we do not require that a quantifier domain is non-empty and we do not require that each individual exists in some domain. Most combinations of these requirements can be found in the literature. One motivation for the choices made here is that they make the translation into two-sorted first-order logic very straightforward, see Subsection 2.10.

DEFINITION 8. Given a varying domain model $\mathfrak{M} = \langle W, R, D, \{\delta_w\}_{w\in W}, \{V_w\}_{w\in W}\rangle$, the relation $\mathfrak{M}, g, w \models \phi$ is defined as in the increasing domain case, that is, Definition 6. The definition of validity is the same as in the constant domain case, that is, Definition 2.

Note that validity is defined as in the constant domain case, not as in the increasing domain case. The resulting difference is that in the varying domain case all assignments are considered whereas in the increasing domain case the only assignments considered are assignments where every variable is assigned an existent. The reason why we make use of different definitions of validity is that straightforward and simple axiom systems are available with these choices.

2.7 Axiom systems for varying domains

In this subsection we shall give two different Hilbert-style axiom systems for the varying domain basic first-order modal logic. First note that the $\forall$ Elimination formula $\forall x\phi \rightarrow \phi[y/x]$ it is not valid with respect to the varying domain semantics (if the variable y designates a non-existent object, then the object is not a member of the domain of the quantifier, so the antecedent can be true but the succedent false) but it is valid with respect to the constant and increasing domain semantics (in both cases the object designated by y is a member of the quantifier domain, so if the antecedent is true, then the succedent is also true). Indeed, the $\forall$ Elimination formula is taken as an axiom in both the constant domain axiom system of Subsection 2.3 and the increasing domain axiom system of Subsection 2.5. Now, the $\forall$ Elimination formula can straightforwardly be modified in two ways such that it becomes valid with respect to the varying domain semantics. It is these two different modified versions of $\forall$ Elimination that give rise to the two varying domain axiom systems we shall give below.

Before giving the first axiom system we need to give a small definition: The so-called existence predicate is defined by the convention that $E(y)$ is an abbreviation for $\exists z(z = y)$ where z is a variable distinct from the variable y. We shall come back to the existence predicate in the next subsection. The first axiom system is obtained by extending the axiom system for the propositional modal logic **K** (see Subsection 2.3) with the axiom schemas

(*Reflexivity*)	$x = x$
(*Substitutivity*)	$(x = y \wedge \phi[x/z]) \rightarrow \phi[y/z]$
(*Necessary Distinctness*)	$x \neq y \rightarrow \Box(x \neq y)$
(*Free $\forall$ Elimination*)	$(\forall x\phi \wedge E(y)) \rightarrow \phi[y/x]$

and the rule

$$\frac{(\psi \wedge E(y)) \rightarrow \phi[y/x]}{\psi \rightarrow \forall x\phi}\ (\textit{Free } \forall \textit{ Introduction})$$

where the rule is equipped with the side-condition that the variable y does not occur free in ψ or in $\forall x\phi$. The axiom system is sound and complete with respect to the varying

domains semantics given in the previous subsection, cf. [49]. (Some of the axioms and rules above are formulated differently in [49], but these differences are not of significance here.)

Note that the axiom (*Free* $\forall$ *Elimination*) above is the result of adding a "guard" formula $E(y)$ to the antecedent of the $\forall$ Elimination formula $\forall x\phi \to \phi[y/x]$ such that the antecedent becomes false in the case where the variable y designates a non-existent. Note moreover that the rule (*Free* $\forall$ *Introduction*) above also makes use of such a guard formula. The axiomatic machinery for quantifiers in this axiom system is identical to the standard axiomatic machinery for quantifiers in so-called free logic which is a variant of ordinary first-order logic where quantifiers only range over a subset of the universe (but where variables might designate any member of the universe as in ordinary first-order logic). One original motivation for developing free logic was to avoid the assumption made in ordinary first-order logic that quantifier domains are non-empty as this assumption was found undesirable by a number of philosophers because of the associated "existential commitment". See [7] for more information on free logic.

We shall now give the second, alternative, axiom system for the varying domain basic first-order modal logic. This axiom system does not involve the existence predicate, in fact, the system is for a version of the basic logic without equality. This alternative system is obtained by extending the axiom system for the propositional modal logic **K** (see Subsection 2.3) with the axiom schemas

(*Vacuous* $\forall$)	$\forall x\phi \leftrightarrow \phi$
($\forall$ *Distributivity*)	$\forall x(\phi \to \psi) \to (\forall x\phi \to \forall x\psi)$
($\forall$ *Permutation*)	$\forall y\forall z\phi \to \forall z\forall y\phi$
(*Restricted* $\forall$ *Elimination*)	$\forall y(\forall x\phi \to \phi[y/x])$

and the rule

$$\frac{\phi}{\forall x\phi}\ (\forall\ \textit{Generalisation})$$

where the axiom (*Vacuous* $\forall$) is equipped with the side-condition that the variable x does not occur free in the formula ϕ. The axiom system is sound and complete with respect to the varying domains semantics given in Subsection 2.6, cf. the book [68]. (In the axiom system of [68], the formula $\forall y\forall z(\forall x\phi \to \phi[y/x])$ is taken as an axiom instead of the axioms ($\forall$ *Permutation*) and (*Restricted* $\forall$ *Elimination*) above, but this difference is not of significance here.) One might ask whether the axiom ($\forall$ *Permutation*) really is needed. This turns out to be the case which was pointed out in [34]. See the discussions of this issue in [68] and [41].

Note the way in which the $\forall$ Elimination formula $\forall x\phi \to \phi[y/x]$ has been modified in the system above: The variable y has been quantified over such that it only designates existents. The history of systems in line with this system goes back to [75].

2.8 Existence and quantification

In connection with varying domain models, the existence predicate is defined by the convention that $E(y)$ is an abbreviation for $\exists z(z = y)$ where z is a variable distinct from the variable y. With this definition it is straightforward to check that for any varying domain model $\mathfrak{M}$, any world w, and any assignment g, the relationship $\mathfrak{M}, g, w \models E(x)$ holds if and only if $g(x) \in \delta_w$. Thus, the existence predicate is true of the individual

designated by some variable if and only if the individual in question exists. So with this definition the existence predicate behaves as desired. But if the definition is adopted in connection with constant domain models, then the existence predicate is true of all individuals as expected.

However, in connection with constant domain models, the existence predicate is usually taken to be primitive, rather than defined. This brings us to an important definition.

DEFINITION 9. Given a varying domain model $\mathfrak{M} = \langle W, R, D, \{\delta_w\}_{w\in W}, \{V_w\}_{w\in W}\rangle$, a constant domain model $\mathfrak{M}^* = \langle W, R, D, \{V^*_w\}_{w\in W}\rangle$ for the language extended with a 1-place predicate symbol E is defined by letting V^*_w be the extension of V_w such that $V_w(E) = \delta_w$.

Clearly, the map $(\cdot)^*$ which maps $\mathfrak{M}$ to $\mathfrak{M}^*$ is bijective. Thus, from a mathematical point of view, giving a varying domain model is the same as extending the language with a 1-place predicate symbol E and then giving a constant domain model.

This observation is exploited in the translation below which translates any formula in the original language (that is, the language without the predicate symbol E) into a formula in the language extended with E.

$$\begin{array}{rcl}
(P(x_1,\dots,x_n))^E & = & P(x_1,\dots,x_n) \\
(x = y)^E & = & x = y \\
(\phi \wedge \psi)^E & = & \phi^E \wedge \psi^E \\
(\neg\phi)^E & = & \neg\phi^E \\
(\Box\phi)^E & = & \Box\phi^E \\
(\forall x\phi)^E & = & \forall x(E(x) \to \phi^E)
\end{array}$$

The translation gives rise to the following result.

PROPOSITION 10. *Let $\mathfrak{M}$ be a varying domain model. For any first-order modal-logical formula ϕ, any world w, and any assignment g, $\mathfrak{M}, g, w \models \phi$ if and only if $\mathfrak{M}^*, g, w \models \phi^E$.*

Proof. Induction in the structure of ϕ. ❑

See [68] and [41] for a more detailed discussion of the existence predicate.

The interpretation of a quantifier in a constant domain model is called *possibilist quantification* since the quantifier ranges over individuals that possibly exist (this terminology is a bit inaccurate if existence is formalised by extending the constant domain semantics with a primitive existence predicate as described above, the reason being that the quantifier then ranges over all individuals, not only those that possibly exist, that is, exist at some world, but we ignore this inaccuracy). On the other hand, the interpretation of a quantifier in a varying domain model is called *actualist quantification* since the quantifier in this case ranges over individuals that actually exist, that is, individuals that exist in the actual world.

The difference between actualist and possibilist quantification is very clear when the modal operator is given a temporal interpretation, that is, when worlds are taken to be instants and the modal operator is interpreted using the earlier-later relation on instants. In that case actualist quantification corresponds to quantifying over things that now exist whereas possibilist quantification corresponds to quantifying over things that exist at some time. This distinction was discussed already by Arthur Prior who rejected the temporal version of possibilist quantification:

> ... even if it be true that whatever exists at any time exists at all times, there is surely no *inconsistency* in denying it, and a *logic* of time-distinctions ought to be able to proceed without assuming it. ([98], p. 30)

Prior equated the statement "x exist" with "there are facts about x" and he found that facts can only be about whatever now exists. This view has the consequence that facts cannot be about things which have ceased to exist. Prior was uncomfortable about this consequence, but he found it unavoidable. Of course, the view also has the consequence that facts cannot be about things which do not yet exist, but Prior considered this less disputable. Since Prior, possibilist and actualist quantification has given rise to much philosophical discussion, see the book [41] for an account. See also the contributions to the discussion given in the papers [97] and [69].

2.9 The Barcan and Converse Barcan formulas

It is straightforward to show that the Barcan formula $\forall x \Box \phi \to \Box \forall x \phi$ is valid in any decreasing domain skeleton, and moreover, it can also be shown straightforwardly that if the Barcan formula is valid in a varying domain skeleton, then the skeleton in question has decreasing domains. Thus, the class of varying domain skeletons that validates the Barcan formula is exactly the class of decreasing domain skeletons. Prior rejected the Barcan formula for the same reasons as he rejected possibilist quantification, see the previous section. It can also be shown straightforwardly that the class of varying domain skeletons that validates the Converse Barcan formula $\Box \forall x \phi \to \forall x \Box \phi$ is exactly the class of increasing domain skeletons. Indeed, as we saw in Subsection 2.5, the Converse Barcan formula is derivable in the axiom system for increasing domains.

First-order modal logics can be seen as combinations of two distinct logics, namely propositional modal logic and ordinary first-order logic. The two logics, propositional modal logic and ordinary first-order logic, are combined in different ways in the constant, increasing, and varying domain logics. The interaction between modality and quantification is stronger in the constant domain logic than in the varying domain logic in the sense that the Barcan and Converse Barcan formulas (which together say that the order of quantifiers and modal operators does not matter) both are valid in the constant domain semantics but none of them are valid in the varying domain semantics. The Barcan formula is not valid in the increasing domain semantics, but the Converse Barcan formula is, so the increasing domain logic has a "medium" interaction between modality and quantification.

The semantical import of the Barcan and Converse Barcan formulas stems from the distinction between the semantics of the formulas $\forall x \Box \phi$ and $\Box \forall x \phi$. This distinction is an example of the so-called *de re*/*de dicto* distinction. In Latin *de re* means "about the thing" and *de dicto* means "about the proposition". To explain this difference, we instantiate the formula ϕ to $P(x)$. The formula $\Box \forall x P(x)$ says that

> it is necessary that each existing thing is P.

This is a *de dicto* interpretation since it says something about a proposition, namely the proposition that each existing thing is P. What it says about this proposition is that it is necessary. On the other hand, the formula $\forall x \Box P(x)$ says that

> each existing thing is necessarily P.

This is a *de re* interpretation since it says something about things, namely the things that exist. What it says about these things is that each of them is necessarily P. Se the book [41] for a much more thorough discussion of *de re* and *de dicto*. We shall return to the *de re*/*de dicto* distinction in Subsection 3.3 where we consider predicate abstraction. The history of formulas like the Barcan and Converse Barcan formulas goes back to the paper [6].

2.10 Translation into two-sorted first-order logic

The basic first-order modal logics can be translated into two-sorted first-order logic with equality. There is one sort for worlds and one sort for individuals. We consider two different translations, one which is truth-preserving with respect to the constant domain semantics and one which is truth-preserving with respect to the varying domain semantics. There is not much literature available on translations of first-order modal logic into sorted first-order logic. The translations we consider in this subsection are variations of a translation given in [127] and they are also considered in the chapter [91] in *Handbook of Automated Reasoning* which moreover considers a range of other non-classical logics. In [127] a semantic characterisation is given of the formulas of two-sorted first-order logic which have the same expressive power as formulas of a variant of the varying domain logic. The papers [63] and [65] consider a number of formulas in two-sorted first-order logic that express properties of models which are not expressible in first-order modal logic. The latter paper concentrates on a first-order version of the modal logic **S5**. A recent example of work in this area is the paper [123] which also concerns the expressive power of a first-order version of **S5**. See also that paper for an overview of the area.

We first consider the constant domain case. The two-sorted first-order language under consideration here is defined as follows. It is assumed that a countably infinite set of first-order variables for worlds and a countably infinite set of first-order variables for individuals are given. The sets are assumed to be disjoint. The metavariables a, b, c, ... range over first-order variables for worlds and the metavariables x, y, z, ... range over first-order variables for individuals. There are no function symbols or constants. Formulas of the two-sorted first-order language are defined by the grammar

$$S \ ::= \quad P^*(a, x_1, \ldots, x_n) \mid R(a, b) \mid x = y \mid S \wedge S \mid \neg S \mid \forall a S \mid \forall x S$$

where P is an n-place predicate symbol of first-order modal logic, a and b are variables for worlds, and x_1, ..., x_n as well as x and y are variables for individuals. Note that according to the grammar above, for each n-place predicate symbol P of the first-order modal language there is a corresponding $(n+1)$-place predicate symbol P^* in the two-sorted first-order language. The two-sorted $(n+1)$-place predicate symbol P^* will be interpreted such that it relativises the interpretation of the corresponding modal n-place predicate symbol P to worlds. In the grammar above R is a designated predicate symbol which will be interpreted using the accessibility relation (with the same name). In what follows, we shall identify first-order variables for individuals with first-order variables of modal logic. Note that the language contains two quantifiers, a quantifier for each sort, but the language only contains one equality predicate, namely an equality predicate for individuals.

We now give the translation. Given two new first-order variables for worlds, a and b, the translations ST_a and ST_b are defined by mutual induction. We just give the

translation ST_a.

$$\begin{array}{rcl} ST_a(P(x_1,\ldots,x_n)) & = & P^*(a,x_1,\ldots,x_n) \\ ST_a(x=y) & = & x=y \\ ST_a(\phi\wedge\psi) & = & ST_a(\phi)\wedge ST_a(\psi) \\ ST_a(\neg\phi) & = & \neg ST_a(\phi) \\ ST_a(\Box\phi) & = & \forall b(R(a,b)\rightarrow ST_b(\phi)) \\ ST_a(\forall x\phi) & = & \forall x ST_a(\phi) \end{array}$$

The definition of ST_b is obtained by exchanging a and b. What has been done is that the semantics of first-order modal logic has been formalised in terms of two-sorted first-order logic, note how each clause in the translation formalizes a clause in the definition of the semantics, Definition 2. The translation is an extension of the well-known *standard translation* from modal logic into first-order logic, see [127]. See Chapter 11 in the present handbook for a temporal version of the translation above.

To state formally that the translation given above is truth-preserving with respect to the constant domain semantics, we make use of the observation that a constant domain model for first-order modal logic can be considered as a model for two-sorted first-order logic and vice versa.

DEFINITION 11. Given a constant domain model $\mathfrak{M}=\langle W,R,D,\{V_w\}_{w\in W}\rangle$ for first-order modal logic, a model $\mathfrak{M}^*=\langle W,D,V^*\rangle$ for two-sorted first-order logic is defined by letting

- $V^*(R)=R$ and
- $(w,d_1,\ldots,d_n)\in V^*(P^*)$ if and only if $(d_1,\ldots,d_n)\in V_w(P)$.

It is straightforward to see that the map $(\cdot)^*$ which maps $\mathfrak{M}$ to $\mathfrak{M}^*$ is bijective. Moreover, if an assignment in the sense of first-order modal logic is extended such that it assigns a world to each first-order variable for worlds, then it can be considered an assignment as appropriate for two-sorted first-order logic and vice versa. See Chapter 11 in the present handbook for a temporal version of the above construction of a two-sorted first-order model from a modal model.

Given a model $\mathfrak{M}$ for two-sorted first-order logic, the relation $\mathfrak{M},g\models\phi$ is defined by induction in the standard way, where g is an assignment for two-sorted first-order logic and ϕ is a two-sorted first-order formula. The formula ϕ is said to be *true* if $\mathfrak{M},g\models\phi$; otherwise it is said to be *false*. By convention $\mathfrak{M}\models\phi$ means $\mathfrak{M},g\models\phi$ for every assignment g. A formula ϕ is valid if and only if $\mathfrak{M}\models\phi$ for any model $\mathfrak{M}$.

We are now ready to state formally that the translation is truth-preserving.

PROPOSITION 12. *Let a constant domain model $\mathfrak{M}$ be given. For any first-order modal-logical formula ϕ and any assignment g for $\mathfrak{M}$, it is the case that $\mathfrak{M},g,g^*(a)\models\phi$ if and only if $\mathfrak{M}^*,g^*\models ST_a(\phi)$ where g^* is any assignment extending g such that it assigns a world to each first-order variable for worlds (and the same for ST_b).*

Proof. Induction in the structure of ϕ. ❑

Thus, first-order modal logic, considered a language for talking about constant domain models, has the same expressive power as the fragment of two-sorted first-order logic obtained by taking the image of first-order modal logic under the translation ST_a.

We now briefly consider the translation which is truth-preserving with respect to the varying domain semantics. The two-sorted first-order language under consideration is the same as in the constant domain case except that the grammar for formulas is extended with a clause $E(a, x)$ for a new predicate symbol where a is a variable for worlds and x is a variable for individuals. Intuitively, the predicate symbol E is interpreted such that it relates a world to individuals existing at that world.

The translation is the same except that the clause for quantifiers is replaced by

$$ST_a(\forall x\phi) \quad = \quad \forall x(E(a, x) \rightarrow ST_a(\phi))$$

It is straightforward to modify Definition 11 to the varying domain case and it is also straightforward to check that Proposition 12 still holds.

Note that the correspondence between varying domain first-order modal models and two-sorted first-order models is very straightforward and simple due to the choices made in Subsection 2.6: We did not require predicates to be false of non-existents, we did not require quantifier domains to be non-empty, and we did not require that each individual exists in some domain.

3 FIRST-ORDER INTENSIONAL LOGIC

The logic we shall consider in this section is more complicated than the basic first-order modal logics we have considered in the previous section. Recall that variables in the basic logics designate rigidly, that is, a variable designates the same object in all worlds. So the assignment of objects to such variables is not relative to worlds. Compared to this, one new piece of machinery of first-order intensional logic is *intension variables* which are variables that designate *intensions*, that is, functions from worlds to objects. Thus, intension variables designate *non-rigidly* in the sense that intensions might map different worlds to different objects. Intensions are also called *individual concepts*. Intension variables can be motivated in a number of different ways. One very instructive motivation is that intension variables allow us to formalise natural language sentences involving non-rigidly designating terms like for example "the number of planets" and "the world champion in marathon running". The first example term designates non-rigidly as it designates the number nine (since there are nine planets in our world), but it might have designated another number (since there might have been another number of planets if natural history had been different). Similarly, the designation of the second example term is the winner of the world championship in marathon running, and the identity of the winning person is obviously also a contingent matter. Intension variables can be used for many different purposes, one example is that they can be used to give a solution to a famous philosophical problem, namely a modal version of Frege's puzzle about the morning star and the evening star. We shall return to this in Subsection 3.2. To sum up, in the previous section we took objects as semantic values, but in the present section we moreover take functions from worlds to objects as semantic values. In fact, first-order intensional logic allows quantification over both objects and intensions as well as predication of both objects and intensions. Thus, predicates are typed, that is, it is specified whether an argument-place is for an object term or an intension term. First-order intensional logic also includes so-called *predicate abstraction* which allows the function which is the interpretation of an intension variable to be applied to an

argument, that is, a world, whereby an object is obtained. Thus, predicate abstraction can be considered the "interface" between intensions and objects. We shall come back to the motivation for predicate abstraction in Subsection 3.3.

A number of versions of first-order intensional logic can be found in the literature. The version we give here is from the paper [40] where also a tableau system can be found which is sound and complete with respect to the semantics. See the book [68], the chapter [49] in *Handbook of Philosophical Logic*, and the papers [108] and [96] for other versions. See also [39] for a treatment of higher-order intensional logic. The history of first-order intensional logic goes back to the work of Richard Montague and Daniel Gallin, see [87] and [48]. See Chapter 21 in the present handbook for a historical account of intensional logic.

3.1 Syntax and semantics of first-order intensional logic

We now extend the formal syntax and semantics of the constant domain basic logic with first-order intensional machinery. Note that constant domains are just as general as varying domains in the sense that the varying domain semantics can be simulated by the constant domain semantics if a primitive existence predicate is added, cf. Subsection 2.8. Conventions and definitions that are the same as in the basic case are not repeated.

First the syntax. It is assumed that a countably infinite set of intension variables is given. The metavariables i, j, k, ... range over intension variables. It is assumed that the set of intension variables is disjoint from the set of ordinary variables for objects. A term is either an ordinary object variable or an intension variable. At this stage function symbols could have been included, but in the interest of simplicity we shall not do so. Predicate symbols are typed, that is, it is not only specified which arity a predicate symbol has, it is also specified which type each argument place has. Following the paper [40], the types of an n-place predicate symbol are specified by a list $T_1, \ldots, T_n$ where $T_i \in \{O, I\}$ for each T_i (the letter O stands for object and the letter I stands for intension). The syntax also includes predicate abstraction as mentioned above. Formulas are defined by the grammar

$$S \ ::= \quad P(t_1, \ldots, t_n) \mid x = y \mid S \wedge S \mid \neg S \mid \Box S \mid \forall x S \mid \forall i S \mid (\lambda x S)(i)$$

where P is an n-place predicate symbol and $t_1, \ldots, t_n$ are terms of the respective types $T_1, \ldots, T_n$ specified for P, x and y are object variables, and i is an intension variable. The free variable occurrences in the predicate abstraction $(\lambda x \phi)(i)$ are the free variable occurrences in the formula ϕ, except for occurrences of x, together with the variable occurrence i. Thus, all occurrences of x in ϕ are bound. Of course, the definition of substitution is modified in accordance with this extension of the language. We allow intension variables to occur in argument places for object variables in the sense that we abbreviate $(\lambda x P(\ldots, x, \ldots))(i)$ as $P(\ldots, i, \ldots)$, etc. This also applies to the equality predicate, so $i = j$ is an abbreviation for $(\lambda y(\lambda x(x = y))(i))(j)$. Now the semantics.

DEFINITION 13. A tuple $\langle W, R, D_O, D_I, \{V_w\}_{w \in W}\rangle$ where

1. W is a non-empty set;

2. R is a binary relation on W;

3. D_O is a non-empty set;

4. D_I is a non-empty set of functions from W to D_O; and
5. for each w, V_w is a function that to each n-place predicate symbol P assigns a subset of $D_{T_1} \times \ldots \times D_{T_n}$ where $T_1, \ldots, T_n$ are the types specified for P.

is a *constant domain intensional model.* The tuple $\langle W, R, D_O, D_I \rangle$ is called a *constant domain intensional skeleton* and the model is said to be *based* on this constant domain intensional skeleton.

Note that both the domain of objects D_O and the domain of intensions D_I are taken to be constant. A version of first-order intensional logic with varying intension domains can be found in [49] (that version also differs in other respects from the version presented here, however).

DEFINITION 14. Let $\mathfrak{M} = \langle W, R, D_O, D_I, \{V_w\}_{w \in W} \rangle$ be a constant domain intensional model. An assignment is a function that to each object variable assigns an element of D_O and to each intension variable assigns an element of D_I. The relation $\mathfrak{M}, g, w \models \phi$ is defined in the same way as in the basic constant domain case, that is, Definition 2, except that the clause for ordinary predicates is replaced by

$$\mathfrak{M}, g, w \models P(t_1, \ldots, t_n) \quad \text{iff} \quad (g(t_1), \ldots, g(t_n)) \in V_w(P)$$

and the following clauses

$$\begin{array}{lll} \mathfrak{M}, g, w \models \forall i \phi & \text{iff} & \text{for any } g' \overset{i}{\sim} g,\ \mathfrak{M}, g', w \models \phi \\ \mathfrak{M}, g, w \models (\lambda x \phi)(i) & \text{iff} & \mathfrak{M}, g', w \models \phi \text{ where } g' \overset{x}{\sim} g \text{ and } g'(x) = g(i)(w) \end{array}$$

for intensional quantification and predicate abstraction are added. Also the definition of validity is the same as in the basic constant domain case.

It is instructive to take a look at a couple of natural language sentences that can be formalised using intension variables. The examples involve the term "the President of the United States" which clearly designates non-rigidly. Consider the sentence

The President of the United States is a Republican.

It says something about the person who is the President of the United States, namely that the person in question is a Republican. If the intension variable i stands for “the President of the United States” and the objectual 1-place predicate symbol Q stands for the predicate “is a Republican”, then the formula $Q(i)$ formalises the statement (where $Q(i)$ is an abbreviation for $(\lambda x Q(x))(i)$). Formally, $Q(i)$ is true at a world w if and only if the designation of i at w belongs to the extension of the predicate Q at w, that is, if and only if the extension of Q at w contains the object obtained by applying the intension designated by i to w. On the other hand, consider the sentence

The President of the United States is an important concept in politics.

This sentence is not about the person who happens to be the President of the United States, rather it is about the concept of the President of the United States. What the sentence says about this concept, is that it is politically important. If the intensional 1-place predicate symbol R stands for the predicate “is a politically important concept”, then the formula $R(i)$ formalises the statement in question. Formally, $R(i)$ is true at

a world w if and only if the extension of R at w contains the intension, that is, the function, designated by i. Clearly, the statement is true, but if "the President of the United States" is replaced by for example "the world champion in marathon running", then it becomes false.

According to the clause above for predicate abstraction, the formula $(\lambda x\phi)(i)$ is true at a world w if and only ϕ is true at w when the variable x is assigned the object obtained by applying the intension designated by i to w. Note that from a mathematical point of view, an intension is just a relation between the sets W and D_O of a particular kind, namely what we usually call the graph of a function, and given such a relation together with an element of W, predicate abstraction is the only built-in machinery in the logic that allows us to perform the mathematical operation we usually call applying a function to an argument, thereby obtaining an element of D_O. Incidentally, in [68] a result is proved according to which a formula in the language of the basic first-order modal logic (see Subsection 2.1) is valid in the basic constant domain semantics (see Subsection 2.2) if and only if the formula is valid in a variant of the intensional semantics given above where all variables designate intensions and all predicates (including the equality predicate) are intensional. The point here is that in such an intensional logic there is no machinery to apply intensions to worlds, that is, there is no machinery that can detect that arguments to predicates have a paarticular inner structure, thus, from a mathematical point of view it does not matter whether the intensional semantics or the basic constant domain semantics is chosen (although the choice of semantics may be of philosophical or metaphysical significance, as pointed out in [68]).

Note that the intensional quantifiers in the semantics above range over elements of the set D_I which is an arbitrary non-empty subset of the set of all functions from W to D_O. An alternative semantics can be obtained by letting D_I be the set of all functions from W to D_O. Contrary to the original semantics, this alternative semantics validates the formula $\Box\exists x P(x) \rightarrow \exists i \Box P(i)$ (note that the 1-place predicate symbol P is objectual, so $P(i)$ is an abbreviation for $(\lambda x P(x))(i)$). Roughly, this formula says that if an object is associated with each accessible world, then there exists an intension which maps each accessible world to the object associated with it. A criticism often raised against this property of being able to make an intension out of any association of objects with worlds is that the choices of objects in such an intension need not in any sense be coherent, contrary to what is intuitively expected. In general, logics along the lines of the alternative logic are unaxiomatisable (although it should be mentioned that no proof is available of unaxiomatisability of the alternative logic described here). See [68] and [49] for proofs of unaxiomatisability of other such logics.

It can be remarked that predicate abstraction plays a role in the modal version of Herbrand's theorem given in the paper [37]. The logic under consideration there is essentially the increasing domain logic of Subsection 2.4 extended with non-rigid constant and function symbols as well as predicate abstraction. The role of predicate abstraction is to enable appropriate Skolemisation of formulas involving modal operators, for example, the formula $\Box\exists x P(x)$ is in the terminology of first-order intensional logic Skolemised as $\Box(\lambda x P(x))(i)$ (abbreviated $\Box P(i)$). In the case of ordinary first-order logic, Herbrand's theorem gives rise to a semi-decision procedure by a reduction to the search for a tautology in a countably infinite set of propositional formulas. A similar result can be proved in the modal case, see [37] and also [38]. We shall come back to predicate abstraction in Subsection 3.3.

3.2 Equality and intensions

Equality in first-order modal logic has given rise to a heated philosophical debate. This debate was initiated by a series of papers where W.V.O. Quine criticised quantified modal logic, see for example [101]. See also Chapter 21 in the present handbook for an account of Quine's criticism. Central in the debate initiated by Quine's papers is the issue of substitution of equals for equals in modal contexts. This is not the place to enter into a detailed philosophical discussion of the problem involved in substitution of equals for equals, so we only give a brief sketch of the problem, and we also only give a brief sketch of how a solution to the problem can be given using intensional variables. See the book [41] for a detailed account of the discussion. Now, consider the statement

> If the morning star is identical to the evening star, then it is necessary that the morning star is identical to the evening star.

which is a modal version of Frege's famous puzzle. This statement is naturally taken to be false (the morning star is the same celestial body as the evening star but this is a contingent fact). How can this statement be formalised in the basic first-order modal logic given in the previous section? An obvious candidate is the formula $x = y \to \Box(x = y)$ where the variables x and y respectively stand for the terms "the morning star" and "the evening star". But this does not work since this formula is valid (whether we take the basic varying domain semantics or the basic constant domain semantics).

Given that the equality predicate is objectual, the diagnosis of the problem is that the variables x and y designate rigidly whereas the terms "the morning star" and "the evening star" designate non-rigidly. Therefore the solution to the problem is to replace the object variables x and y by intension variables i and j since intension variables designate non-rigidly. The resulting formula $i = j \to \Box(i = j)$ is not valid, as i and j designating the same object at a world w does not imply that i and j designate the same object in any world accessible from w. Thus, the significant difference is that object variables designate rigidly whereas intension variables designate non-rigidly.

The fact that the formula $i = j \to \Box(i = j)$ is not valid shows that we cannot substitute equals for equals in modal contexts as far as intension variables are concerned. In fact, the failure of substitution of equals for equals in modal contexts is often taken as a criteria for identifying intensional terms. However, note that the formula is valid if objectual equality is replaced by intensional equality, also called *synonymy*, which takes two intensions to be equal if and only if they have the same graph.

To sum up, the formula $x = y \to \Box(x = y)$ is valid as it is, but it is invalid if the rigidly designating object variables x and y are replaced by the non-rigidly designating intension variables i and j. Thereby a solution can be given to the problem of formalizing the modal version of Frege's puzzle. Another solution is to keep the language as it is, but instead generalise the models for the basic first-order modal logic to encompass so-called counterpart relations. This also makes the formula invalid. The history of counterpart relations goes back to the papers [79] and [80] by David Lewis. After the publication of these papers, a number of generalised versions of Lewis' counterpart semantics have been introduced, one example being the semantics given in the paper [72]. See the discussion in the paper [40] where first-order intensional logic is compared to Lewis' counterpart semantics as well as to a variation of the semantics given in [72]. Another formalization of Lewis's counterpart semantics is the semantics considered in Section 11 of Part II of this chapter.

3.3 Predicate abstraction

The motivation for predicate abstraction is closely related to the *de re/de dicto* distinction described in Subsection 2.9. The history of predicate abstraction goes back to the papers [122] and [126]. See also the paper [35]. See [41] for a recent treatment of predicate abstraction. Many natural language sentences are ambiguous as they can be given two distinct readings, a *de re* reading and a *de dicto* reading. Predicate abstraction can be used to distinguish formally between such readings. Consider for example the sentence

> The number of planets is necessarily greater than five.

which is taken from Quine's paper [101]. On one reading, this sentence says that

> it is necessary that the number of planets is greater than five.

This is the *de dicto* reading since it says something about a proposition, namely the proposition that the number of planets is greater than five. It says about this proposition that it is necessary. However, on another reading, the sentence says that

> the number designated by the term "the number of planets" is necessarily greater than five.

This is the *de re* reading since it says something about a thing, namely a number. It says about this number that it is necessarily greater than five. Note that the *de re* reading of Quine's example sentence is naturally taken to be true (since there are nine planets and the number nine is necessarily greater than five) whereas the *de dicto* reading is naturally taken to be false (since there might have been five planets or fewer if natural history had been different). The point here is that the term "the number of planets" designates non-rigidly.

In what follows, the intension variable i stands for "the number of planets" and the objectual 1-place predicate symbol P stands for the predicate "is greater than five". (The term "the number of planets" is actually a so-called definite description. We ignore this additional structure since it is not of significance for the discussion here, but we remark that the term alternatively could have been formalised by a definite description operator, see [41].) The formula $\Box P(i)$ (which is an abbreviation for $\Box(\lambda x P(x))(i)$ since P is objectual) then formalises the *de dicto* reading of Quine's sentence since this formula expresses that

> it is necessary that the thing designated by i is P.

That is, it says something about the proposition that the thing designated by i is P, namely that this proposition is necessary. Formally, $\Box P(i)$ is true at a world w if and only if for each world v accessible from w, the designation of i at v belongs to the extension of the predicate P at v. Thus, in the *de dicto* case the predicate P and its argument, the variable i, are interpreted at the same world, namely the new world v. How about the *de re* reading of Quine's sentence? We want a formula which expresses that

> the thing designated by i is necessarily P.

That is, we want a formula which says something about the thing that i designates, namely that it is necessarily P. So, formally we want the variable i to be interpreted

at the original world w, not at the new world v where the predicate P is interpreted. It is straightforward that the formula $(\lambda x \Box P(x))(i)$ does the job since it is true at the world w exactly under the condition we want, namely under the condition that the designation of i at the world w belongs to the extension of the predicate P at each world v accessible from w. We have used predicate abstraction to indicate that the variable i has to be interpreted at w, not v (note that this is a formally significant difference exactly because we allow non-rigid designation, that is, the interpretations of i at the worlds w and v might not be the same). Thus, predicate abstraction enables us to separate the interpretation of a predicate from the interpretation of its arguments.

3.4 Translation of first-order intensional logic

In Subsection 2.10 it was shown that the basic constant domain logic can be translated into two-sorted first-order logic. In a similar way first-order intensional logic can be translated into three-sorted first-order logic with equality. There is one sort for worlds, one sort for objects, and one sort for intensions. It should be mentioned that the material presented in this subsection has not been presented elsewhere.

The three-sorted first-order language under consideration here is defined as follows. It is assumed that countably infinite sets of first-order variables for respectively worlds, objects, and intensions are given. The three sets are assumed to be pairwise disjoint. As in Subsection 2.10, the metavariables a, b, c, ... range over variables for worlds, and x, y, z, ... range over variables for objects. The metavariables i, j, k, ... range over variables for intensions. There is only one function symbol, namely the 2-place function symbol ℓ which is of type objects and whose argument places are of types intensions and worlds respectively. Thus, a term for worlds is a variable, a term for intensions is a variable, and a term for objects is either a variable or of the form $\ell(i,a)$ where i is a variable for intensions and a is a variable for worlds. Formulas of the three-sorted first-order language are defined by the grammar

$$S \ ::= \quad P^*(a,t_1,\ldots,t_n) \mid R(a,b) \mid t = u \mid i = j \mid S \wedge S \mid \neg S \mid \forall a S \mid \forall x S \mid \forall i S$$

where P is an n-place predicate symbol of first-order intensional logic and t_1, ..., t_n are terms of the respective types T_1, ..., T_n specified for P, a and b are variables for worlds, t and u are terms for objects, i and j are variables for intensions, and x is a variable for objects. As in Subsection 2.10, we identify first-order variables for objects with object variables of modal logic. Similarly, we identify first-order variables for intensions with intension variables of modal logic.

We now give the translation. The translation is obtained by modifying the constant domain version of ST_a given in Subsection 2.10 by replacing the clause for ordinary predicates by

$$ST_a(P(t_1,\ldots,t_n)) \quad = \quad P^*(a,t_1,\ldots,t_n)$$

and by adding the clauses

$$\begin{aligned} ST_a(\forall i \phi) \quad &= \quad \forall i ST_a(\phi) \\ ST_a((\lambda x \phi)(i)) \quad &= \quad ST_a(\phi)[\ell(i,a)/x] \end{aligned}$$

for intensional quantification and predicate abstraction. The translation ST_b is modified in the same way.

In Subsection 2.10 we had a bijective correspondence between models for first-order modal logic and models for two-sorted first-order logic. We do not have a bijective correspondence between models for first-order intensional logic and models for three-sorted first-order logic in general, but we do have a bijective correspondence if a very natural class of models for three-sorted first-order logic is considered instead of the class of all such models. Below we shall make this more precise.

DEFINITION 15. Let $\mathfrak{M} = \langle W, R, D_O, D_I, \{V_w\}_{w \in W}\rangle$ be a constant domain intensional model. A three-sorted first-order model $\mathfrak{M}^* = \langle W, D_O, D_I, V^*\rangle$ is defined by letting

- $V^*(R) = R$,
- $(w, d_1, \ldots, d_n) \in V^*(P^*)$ if and only if $(d_1, \ldots, d_n) \in V_w(P)$, and
- $V^*(\ell)(f, w) = f(w)$.

Thus, the construction of a three-sorted first-order model is straightforward: We use the recipe from Definition 11, and besides that, we take the domain of intensions D_I as it is and we interpret the function symbol ℓ as the application function. Clearly, the map $(\cdot)^*$ is injective.

DEFINITION 16. Let $\mathfrak{M} = \langle W, D_O, D_I, V\rangle$ be a three-sorted first-order model which satisfies the condition that $\mathfrak{M} \models \forall i \forall j(\forall a(\ell(i,a) = \ell(j,a)) \rightarrow i = j)$. A function σ from D_I to the set of functions from W to D_O is defined by letting $\sigma(d)(w) = V(\ell)(d, w)$ (note that σ is injective and therefore has an inverse σ^{-1} on the image $\sigma(D_I)$ of D_I under σ). A constant domain intensional model $\mathfrak{M}^\sharp = \langle W, R^\sharp, D_O, D_I^\sharp, \{V_w^\sharp\}_{w \in W}\rangle$ is defined by letting

- $R^\sharp = V(R)$,
- $D_I^\sharp = \sigma(D_I)$, and
- $(d_1, \ldots, d_n) \in V_w^\sharp(P)$ if and only if $(w, d_1', \ldots, d_n') \in V(P^*)$.

where in the last item $d_i' = d_i$ if $T_i = O$ and $d_i' = \sigma^{-1}(d_i)$ if $T_i = I$ (recall that $T_1, \ldots, T_n$ are the types specified for P). Moreover, given an assignment g for $\mathfrak{M}$, an assignment $g^\sharp$ for $\mathfrak{M}^\sharp$ is defined by letting $g^\sharp(x) = g(x)$ for any object variable x and by letting $g^\sharp(i) = \sigma(g(i))$ for any intension variable i (note that the values of the assignment g on world variables are ignored).

So the construction of a constant domain intensional model is also straightforward: We define $R^\sharp$ in the obvious way and we take the domain of intensions $D_I^\sharp$ to contain any function f from worlds to objects that is "encoded" by some element d of D_I in the sense that f is identical to the function that maps a world w to $V(\ell)(d, w)$. Moreover, we use the bijection σ to move forwards and backwards between D_I and $D_I^\sharp$. The condition $\mathfrak{M} \models \forall i \forall j(\forall a(\ell(i,a) = \ell(j,a)) \rightarrow i = j)$ ensures that different elements of D_I encode different functions from worlds to objects, so the code of an encoded function is uniquely determined. Also, note that $D_I^\sharp$ is non-empty since D_I is non-empty. The map $(\cdot)^\sharp$ is not injective since there is no restriction on the way in which an element of D_I encodes a function from worlds to objects, in particular, an element of D_I need not be identical to the function from worlds to objects that it encodes. However, if D_I is a set of functions

from W to D_O and each such function encodes itself, that is, $V(\ell)$ is the application function, then $(\cdot)^\sharp$ clearly is injective.

The mapping $\mathfrak{M}^*$ of any constant domain intensional model $\mathfrak{M}$ satisfies the condition of Definition 16, and moreover, it is straightforward that $(\mathfrak{M}^*)^\sharp = \mathfrak{M}$. It follows that $((\mathfrak{M}^*)^\sharp)^* = \mathfrak{M}^*$. So if the map $(\cdot)^\sharp$ is restricted to the image of the class of all first-order intensional models under the map $(\cdot)^*$, which of course is the class of all models where D_I is a set of functions from W to D_O and $V(\ell)$ is the application function, then the maps $(\cdot)^*$ and $(\cdot)^\sharp$ are each others' inverses.

Now, given a model $\mathfrak{M}$ for three-sorted first-order logic, the relation $\mathfrak{M}, g \models \phi$ is defined by induction in the standard way, where g is an assignment for three-sorted first-order logic and ϕ is a three-sorted first-order formula. This leads to two propositions. The first concerns the map $(\cdot)^*$.

PROPOSITION 17. *Let $\mathfrak{M}$ be a constant domain intensional model. For any formula ϕ of first-order intensional logic and any assignment g for $\mathfrak{M}$, it is the case that $\mathfrak{M}, g, g^*(a) \models \phi$ if and only if $\mathfrak{M}^*, g^* \models ST_a(\phi)$ where g^* is any assignment extending g such that it assigns a world to each first-order variable for worlds (and the same for ST_b).*

Proof. Induction in the structure of ϕ. ❑

The second proposition concerns the map $(\cdot)^\sharp$.

PROPOSITION 18. *Let $\mathfrak{M}$ be a three-sorted first-order model having the property that $\mathfrak{M} \models \forall i \forall j (\forall a(\ell(i,a) = \ell(j,a)) \to i = j)$. For any formula ϕ of first-order intensional logic and any assignment g for $\mathfrak{M}$, it is the case that $\mathfrak{M}^\sharp, g^\sharp, g(a) \models \phi$ if and only if $\mathfrak{M}, g \models ST_a(\phi)$ (and the same for ST_b).*

Proof. Induction in the structure of ϕ. ❑

We are now ready to prove that validity in first-order intensional logic can be simulated by validity in three-sorted first-order logic.

THEOREM 19. *Any formula ϕ of first-order intensional logic is valid if and only if the formula*

$$\forall i \forall j (\forall a(\ell(i,a) = \ell(j,a)) \to i = j) \to ST_a(\phi)$$

of three-sorted first-order logic is valid.

Proof. By Proposition 17 and Proposition 18. ❑

4 FIRST-ORDER HYBRID LOGIC

First-order hybrid logic is obtained by adding to first-order modal logic further expressive power in the form of a new sort of propositional symbol called a *nominal*, and moreover, by adding so-called *satisfaction operators*. It is stipulated that a nominal is true at exactly one world, so in this sense a nominal refers to a world. If a is a nominal and ϕ is an arbitrary formula, then a new formula $a : \phi$ called a *satisfaction statement* can be formed. The part $a :$ of the satisfaction statement $a : \phi$ is called a satisfaction operator. The satisfaction statement $a : \phi$ expresses that the formula ϕ is true at one particular world,

namely the world at which the nominal a is true. Furthermore, the so-called *binders* $\forall$ and $\downarrow$ might be added. The two binders bind nominals to worlds in two different ways: The binder $\downarrow$ binds a nominal to the actual world whereas the binder $\forall$ quantifies over worlds (but it is not to be confused with the first-order quantifier $\forall$ that has been used earlier in the chapter). The $\downarrow$ binder is definable in terms of $\forall$. Here we shall concentrate on the $\downarrow$ binder.

The history of hybrid logic goes back to Prior's work, more precisely, it goes back to what he called four grades of tense-logical involvement. They were presented in the book [100], Chapter XI (also Chapter XI in the new edition [62]). See also [99] Chapter V.6 and Appendix B.3–4. The stages progress from what can be regarded as pure first-order earlier-later logic to what can be regarded as pure tense logic; the goal being to be able to consider the tense logic of the fourth stage as encompassing the earlier-later logic of the first stage. In other words, the goal was to be able to translate the first-order logic of the earlier-later relation into tense logic. With this in mind, Prior introduced so-called instant-propositions:

> What I shall call the third grade of tense-logical involvement consists in treating the instant-variables a, b, c, etc. as also representing propositions. ([100], p. 122–123)

In the context of modal logic, Prior called such propositions possible-world-propositions. Of course, this is what we here call nominals. Prior also introduced the binder $\forall$ and what we here call satisfaction operators. See the paper [92] and the handbook chapter [93] for accounts of Prior's work. Moreover, see the very recent paper [10] as well as the book [21].

It is notable that hybridisation of propositional as well as first-order modal logics enables the formulation of uniform proof-rules for wide classes of logics. See the papers [9] and [12] for tableau systems and see the papers [16] and [17] for natural deduction systems. The classes of logics considered in [17] correspond to first-order conditions on the accessibility relations and quantifier domains expressed by so-called *geometric theories*. Natural deduction systems corresponding to different geometric theories are obtained in a uniform way simply by adding inference rules as appropriate. It is also notable that first-order hybrid logic offers precisely the features needed to prove interpolation theorems: While interpolation fails in a number of well-known first-order modal logics, see [32], their hybridised counterparts have this property, see the papers [3] and [13]. See Chapter 14 of the present handbook for a detailed introduction to hybrid logic.

4.1 Syntax and semantics of first-order hybrid logic

We now extend the formal syntax and semantics of first-order modal logic with hybrid machinery. We hybridise the varying domain basic logic given in Section 2.6 since proof procedures are available for this logic, namely the tableau and natural deduction systems of respectively [12] and [17] mentioned above. Moreover, an axiom system is available which we shall cover in Section 4.2.

First the syntax. It is assumed that a countably infinite set of nominals is given. The metavariables a, b, c, ... range over nominals. Note that nominals are the only sort of propositional symbols, since ordinary propositional symbols are represented by 0-place predicate symbols. We also add satisfaction operators and the binder $\downarrow$ as mentioned

above. We furthermore assume that a set of non-rigid constant symbols is given, and we follow [12] in overloading the notation for the satisfaction operator by defining a term to be either a first-order variable or an expression of the form $a : f$ where a is a nominal and f is a non-rigid constant symbol. Of course, the term $a : f$ denotes the value of f at the world where a is true. Such terms are called *grounded definite descriptions*. The formulas of first-order hybrid logic are defined by the grammar

$$S \ ::= \quad P(t_1, \ldots, t_n) \mid t = u \mid a \mid S \wedge S \mid \neg S \mid \Box S \mid a : S \mid \forall x S \mid \ \downarrow a S$$

where P is an n-place predicate symbol, $t_1, \ldots, t_n$ as well as t and u are terms, a is a nominal, and x is an ordinary first-order variable. The free nominal occurrences in the formula $a : \phi$ is the occurrence of a together with the free nominal occurrences in ϕ. The free nominal occurrences in $\downarrow a\phi$ are the free nominal occurrences in ϕ, except for occurrences of a. Substitution of nominals for nominals is defined accordingly. Substitution of terms for first-order variables is modified to take into account that terms might contain nominals (that can be bound). Now the semantics.

DEFINITION 20. A *varying domain hybrid model* is a varying domain model as defined in Definition 7, that is, a tuple, $\langle W, R, D, \{\delta_w\}_{w\in W}, \{V_w\}_{w\in W}\rangle$, where for each w, the valuation V_w is extended such that to each non-rigid constant symbol it assigns an element of D.

Thus, hybridisation does not change the notion of a varying domain model except that interpretations of the non-rigid constants are added.

DEFINITION 21. Given a model $\mathfrak{M} = \langle W, R, D, \{\delta_w\}_{w\in W}, \{V_w\}_{w\in W}\rangle$, an assignment is a function that to each first-order variable assigns an element of D and to each nominal assigns an element of W. Given an assignment g, each term t is assigned an element $t^{\mathfrak{M},g}$ of D as follows: If t is of the form $a : f$, then $t^{\mathfrak{M},g} = V_{g(a)}(f)$, otherwise t is a variable, in which case $t^{\mathfrak{M},g} = g(t)$. The relation $\mathfrak{M}, g, w \models \phi$ is defined in the same way as in the basic varying domain case, that is, Definition 8, except that the clauses for predicates are replaced by

$$\begin{array}{rcl} \mathfrak{M}, g, w \models P(t_1, \ldots, t_n) & \text{iff} & (t_1^{\mathfrak{M},g}, \ldots, t_n^{\mathfrak{M},g}) \in V_w(P) \\ \mathfrak{M}, g, w \models t = u & \text{iff} & t^{\mathfrak{M},g} = u^{\mathfrak{M},g} \end{array}$$

and clauses

$$\begin{array}{rcl} \mathfrak{M}, g, w \models a & \text{iff} & w = g(a) \\ \mathfrak{M}, g, w \models a : \phi & \text{iff} & \mathfrak{M}, g, g(a) \models \phi \\ \mathfrak{M}, g, w \models \downarrow a\phi & \text{iff} & \mathfrak{M}, g', w \models \phi \text{ where } g' \stackrel{a}{\sim} g \text{ and } g'(a) = w \end{array}$$

for hybrid machinery are added. Also the definition of validity is the same as in the basic varying domain case.

Propositional hybrid logic has a number of notable features: We can express that a formula ϕ is true at a world a (by the formula $a : \phi$), that a world a is identical to a world c (by the formula $a : c$), and that a world a is R-related to a world c (by the formula $a : \Diamond c$). In the first-order case we can moreover express that an individual t exists at a world a (by the formula $a : E(t)$). These features are exactly what enable the formulation of uniform natural deduction rules for the class of first-order hybrid logics corresponding to conditions expressed by geometric theories, cf. the paper [17].

Grounded definite descriptions can be motivated by the fact that they give an alternative way to formalise the two distinct readings of Quine's example sentence considered in Subsection 3.3. Recall that the *de dicto* reading of the sentence in question was formalised as the formula $\Box(\lambda x P(x))(i)$ in first-order intensional logic whereas the *de re* reading was formalised as $(\lambda x \Box P(x))(i)$. If the non-rigid designator f is considered instead of the intension variable i, then the two readings can be formalised in first-order hybrid logic as respectively $\Box \downarrow a P(a : f)$ and $\downarrow a \Box P(a : f)$. This is no coincidence: If first-order intensional logic is extended with the hybrid-logical machinery of this section and non-rigid designators are replaced by intension variables, then predicate abstractions are eliminable since a formula $(\lambda x \phi)(i)$ is equivalent to $\downarrow a \phi[a : i/x]$ where the nominal a is new.

4.2 Axioms for first-order hybrid logic

In this subsection we give a Hilbert-style axiom system for first-order hybrid logic. The axioms of the system are all substitution-instances of tautologies of propositional logic together with all substitution-instances of the following axiom schemas

(: *Distributivity*)	$a : (\phi \to \psi) \leftrightarrow (a : \phi \to a : \psi)$
(*Falsum*)	$a : \bot \to \bot$
(*Scope*)	$a : b : \phi \leftrightarrow b : \phi$
(*Reflexivity 1*)	$a : a$
(*Reflexivity 2*)	$t = t$
(*Transfer*)	$a : (t = u) \to c : (t = u)$
(: *Introduction*)	$(a \wedge \phi) \to a : \phi$
(*Nominal*)	$a : c \to (a : q) = (c : q)$
(*Substitutivity*)	$(t = u \wedge \phi[t/x]) \to \phi[u/x]$
($\Box$ *Elimination*)	$(\Box\phi \wedge \Diamond e) \to e : \phi$
(*Free* $\forall$ *Elimination*)	$(\forall x \phi \wedge E(t)) \to \phi[t/x]$
($\downarrow$ *Elimination*)	$(\downarrow b \phi \wedge e) \to e : \phi[e/b]$

The rules of the system are the following

$$\frac{\phi \to \psi \qquad \phi}{\psi}\ (\textit{Modus Ponens}) \qquad\qquad \frac{\phi}{a : \phi}\ (:\ \textit{Necessitation})$$

$$\frac{a : \phi}{\phi}\ (\textit{Naming}) \qquad\qquad \frac{(\psi \wedge \Diamond c) \to c : \phi}{\psi \to \Box\phi}\ (\Box\ \textit{Introduction})$$

$$\frac{(\psi \wedge E(y)) \to \phi[y/x]}{\psi \to \forall x \phi}\ (\textit{Free}\ \forall\ \textit{Introduction}) \qquad\qquad \frac{(\psi \wedge c) \to c : \phi[c/b]}{\psi \to \downarrow b \phi}\ (\downarrow\ \textit{Introduction})$$

where the rule (*Naming*) is equipped with the side-condition that the nominal a does not occur in the formula ϕ, the rule (*Free* $\forall$ *Introduction*) is equipped with the side-condition that y does not occur free in $\forall x \phi$ or ψ, the rule ($\Box$ *Introduction*) is equipped with the side-condition that c does not occur free in ϕ or ψ, and the rule ($\downarrow$ *Introduction*) is equipped with the side-condition that c does not occur free in $\downarrow b \phi$ or ψ. The axiom system is sound and complete with respect to the semantics given in Subsection 4.1. A

completeness proof can be found in the paper [17]. The axiom system can be extended with rules corresponding to geometric theories, again see [17]. The system is an extension of an axiom system for propositional hybrid logic given in the paper [18] where also further references on axiom systems for hybrid logic can be found.

Note that the axiom (*Free* $\forall$ *Elimination*) and the rule (*Free* $\forall$ *Introduction*) above are the same as the axiom and the rule for quantifiers of Subsection 2.7. It is instructive to compare these rules to the axiom ($\Box$ *Elimination*) and the rule ($\Box$ *Introduction*) for the modal operator. As described in Subsection 2.7, the idea in the rule (*Free* $\forall$ *Introduction*) is that the guard formula $E(y)$ in the antecedent ensures that the antecedent is false in the case where the variable y refers to an individual outside the range of the quantifier. This is analogous to the idea in the rule ($\Box$ *Introduction*) for hybrid logic which is that the guard formula $\Diamond c$ in the antecedent ensures that the antecedent is false in the case where the nominal c refers a world that is not accessible. A similar remark applies in connection with the pair of rules (*Free* $\forall$ *Elimination*) and ($\Box$ *Elimination*). In fact, such analogies can be found in connection with all pairs of Elimination rules and all pairs of Introduction rules.

4.3 Translation of first-order hybrid logic

First-order hybrid logic can be translated into two-sorted first-order logic by an extension of the varying domain version of the translation of Subsection 2.10.

The two-sorted first-order language under consideration here is a straightforward modification of the varying domain version of the language of Subsection 2.10. A term for worlds is still a variable but now a term for individuals is either a variable or of the form $f(a)$ where f is a constant symbol of first-order hybrid logic. Formulas of the language are defined by the grammar

$$S \ ::= \quad P^*(a, t_1, \ldots, t_n) \mid R(a, b) \mid a = b \mid t = u \mid E(a, t) \mid S \wedge S \mid \neg S \mid \forall a S \mid \forall x S$$

where P is an n-place predicate symbol of first-order modal logic, a and b are variables for worlds, and $t_1, \ldots, t_n$ as well as t and u are terms for individuals. Note that the clause $a = b$ has been added. We shall identify first-order variables for worlds with nominals in the same way as we have identified first-order variables for individuals with first-order variables of modal logic.

A term t of first-order hybrid logic is translated by the translation ST defined as follows: If t is of the form $a : f$, then $ST(t) = f(a)$, otherwise t is a variable, in which case $ST(t) = t$. Note that the translation ST of terms is not relative to a variable for worlds. A formula is translated by a translation obtained by modifying the varying domain version of ST_a given in Subsection 2.10 by replacing the clauses for predicates by

$$\begin{array}{rcl} ST_a(P(t_1, \ldots, t_n)) & = & P^*(a, ST(t_1), \ldots, ST(t_n)) \\ ST_a(t = u) & = & ST(t) = ST(u) \end{array}$$

and by adding the clauses

$$\begin{array}{rcl} ST_a(c) & = & a = c \\ ST_a(c : \phi) & = & ST_a(\phi)[c/a] \\ ST_a(\downarrow c\phi) & = & ST_a(\phi)[a/c] \end{array}$$

The translation ST_b is modified in the same way. A similar translation can be found in the paper [3].

It is straightforward to adapt the varying domain version of Definition 11 to the hybrid-logical case by extending it to encompass non-rigid constant symbols, so we still have a bijective correspondence between models for first-order hybrid logic and models for the two-sorted first-order logic under consideration here. Moreover, the notions of assignments are the same. Given this, it is also straightforward to adapt the varying domain version of Proposition 12 to the hybrid-logical case.

It turns out that a fragment of two-sorted first-order logic can be translated back into first-order hybrid logic. This fragment is defined by the grammar

$$\begin{array}{rl} S \ ::= & P^*(a, t_1, \ldots, t_n) \mid R(a, c) \mid E(a, t) \mid a = c \mid t = u \mid S \wedge S \mid \neg S \mid \\ & \forall b(R(a, b) \to S) \mid \forall x(E(a, x) \to S) \end{array}$$

where the variables a and b are distinct. A term t of two-sorted first-order logic is translated back into first-order hybrid logic by the translation HT defined as follows: If t is of the form $f(a)$, then $HT(t) = a : f$, otherwise t is a variable, in which case $HT(t) = t$. So HT and the translation ST given above are simply inverses to each other. A formula is translated by the translation given below.

$$\begin{array}{rcl} HT(P^*(a, t_1, \ldots, t_n)) & = & a : P(HT(t_1), \ldots, HT(t_n)) \\ HT(R(a, c)) & = & a : \Diamond c \\ HT(E(a, t)) & = & a : E(HT(t)) \\ HT(a = c) & = & a : c \\ HT(t = u) & = & HT(t) = HT(u) \\ HT(\phi \wedge \psi) & = & HT(\phi) \wedge HT(\psi) \\ HT(\neg\phi) & = & \neg HT(\phi) \\ HT(\forall b(R(a, b) \to \phi)) & = & a : \Box \downarrow b HT(\phi) \\ HT(\forall x(E(a, x) \to \phi)) & = & a : \forall x HT(\phi) \end{array}$$

The propositional version of this translation was originally given in [2] where the associated fragment of ordinary one-sorted first-order logic is called the *bounded fragment*. In [2] a number of independent semantic characterisations of the bounded fragment are given. See also [3].

The translation HT is truth-preserving as is shown by the proposition below (where $\mathfrak{M}^*$ is the model for two sorted first-order logic defined in the hybrid-logical version of Definition 11).

PROPOSITION 22. *Let a varying domain hybrid model $\mathfrak{M}$ be given. For any formula ϕ of the two-sorted version of the bounded fragment and any assignment g for $\mathfrak{M}$, it is the case that $\mathfrak{M}^*, g \models \phi$ if and only if $\mathfrak{M}, g \models HT(\phi)$.*

Proof. Induction in the structure of ϕ. ❑

Thus, in the sense of the proposition above and the hybrid-logical version of Proposition 12, first-order hybrid logic has the same expressive power as the two-sorted version of the bounded fragment (note that for any formula ϕ of first-order hybrid logic, the formula $ST_a(\phi)$ is in this fragment).

5 OTHER SURVEY ARTICLES AND BOOKS

Below we point out a number of survey articles and books that contain material on first-order modal logic not covered by the present handbook chapter. First the survey articles.

- *Philosophical perspectives on quantification in tense and modal logic*, N.B. Cocchiarella [20], in *Handbook of Philosophical Logic.* Discusses a range of philosophical issues.

- *Basic modal logic*, M. Fitting [36], in *Handbook of Logic in Artificial Intelligence and Logic Programming.* Contains a condensed introduction to first-order modal logic including tableau systems.

- *Quantification in modal logic*, J.W. Garson [49], in *Handbook of Philosophical Logic.* An introduction to first-order modal logic that gives a good overview of the whole area. Contains a detailed discussion of completeness and incompleteness.

- *Correspondence theory*, J. van Benthem [129], in *Handbook of Philosophical Logic.* Contains a section on correspondence theory for first-order modal logic.

Below is the list of books.

- *First-Order Modal Logic*, M. Fitting and R. Mendelsohn [41]. A detailed introduction to first-order modal logic that covers technical as well as philosophical issues. Includes tableau systems.

- *Temporal Logic: Mathematical Foundations and Computational Aspects (Volume 1)*, D. Gabbay and I. Hodkinson and M. Reynolds [43]. Contains a chapter on first-order temporal logic.

- *Quantification in Nonclassical Logic*, D. Gabbay and V. Shehtman and D. Skvortsov [44]. A detailed mathematical treatment of first-order modal logic and other first-order non-classical logics.

- *Modal Logics and Philosophy*, R. Girle [58]. Gives an introduction to first-order modal logic from a philosophical point of view.

- *The Logics of Time and Computation*, R. Goldblatt [59]. Has some material on first-order dynamic logic.

- *Dynamic Logic*, D. Harel, D. Kozen, and J. Tiuryn [61]. Gives a detailed introduction to first-order dynamic logic.

- *A New Introduction to Modal Logic*, G.E. Hughes and M.J. Cresswell [68]. Contains an introduction to first-order modal logic that covers a broad range of topics. Compares intensions and counterparts. A follow-up to the books [67] and [66] by the same authors.

- *Modal Logic and Classical Logic*, J. van Benthem [127]. Contains a chapter on first-order modal logic which includes translational issues.

6 INTRODUCTION TO PART II

In this second part of the chapter, we first deal with more specific topics concerning *decision, completeness and axiomatizability* issues. We shall see in Section 7 that the extension to modal languages of a number of well-known decidability results for fragments of classical logic is hopeless; however, less naive extensions (limiting the kind of subformulas occurring within the scope of modal operators) still keep decidability over classical fragments. This is a remarkable fact, because the expressivity of such combined fragments is indeed quite rich, thanks to the contribution of the modal operators.

Completeness analysis of normal systems over **QS4** will reveal in Section 8 the intrinsic limits of Kripke semantics; in addition, rather natural classes of frames will turn out to be non-axiomatizable. However, unlike undecidability and non-axiomatizability results, incompleteness results cannot properly be seen as negative results: on the contrary, they seem to indicate that modal logic cannot be reduced to possible worlds semantics and that extra motivations for it can be found elsewhere, in alternative (non-Kripkean) semantics.

Such *alternative semantics* will be investigated in the remaining sections, using the hyperdoctrinal point of view as a unifying tool. Modalities arising from geometric morphisms of toposes will be studied in Subsection 10.1 and in Subsection 10.2 we shall exploit the isomorphism between counterpart frames and preordered topological bundles in order to find the relevant hints for the axiomatization of counterpart semantics. The axiomatization of counterpart semantics is presented in Section 11 (this section is independent from the rest of the chapter, the reader can have *direct access* to it after reading only Subsection 9.1).

We summarize here basic syntactic and semantic ingredients, just to fix notation (for more information, consult Section 2 in the First Part of this chapter). We fix a first-order language $\mathcal{L}$ (without identity, functions and constant symbols, for simplicity)[1] containing infinitely many predicate symbols for each arity $n \geq 0$ (a special 0-ary predicate symbol $\perp$ denoting syntactic falsehood is included in $\mathcal{L}$). Formulas are built up using countably many variables, propositional implication $\rightarrow$, the quantifier $\forall$ and the modal connective $\Box$ (the other operators $\top, \wedge, \vee, \neg, \leftrightarrow, \exists, \Diamond$ are defined in the usual way). Notations like $\phi(x_1, \dots, x_n)$ (or, for short, $\phi(\underline{x})$) means that ϕ contain free variables only among the tuple of distinct variables $\underline{x} := x_1, \dots, x_n$.

A *first-order modal system* **S** is a set of formulas closed under necessitation, modus ponens, universal generalization and uniform substitution rules (for the definition of uniform substitution in the predicate case, consult Subsection 2.3 of Part I); since we deal only with normal extensions of **QK**, we assume also that **S** contains, in addition to all classically valid formulas, also the formulas **K**, namely the formulas of the kind $\Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$. If **L** is a propositional modal logic (i.e., a set of propositional formulas closed under modus ponens, necessitation, uniform substitution and containing all classical tautologies and the formulas **K**), we denote by **QL** the minimum first-order modal system containing **L**; by **BF**.**L** we mean the minimum first order modal system containing both **L** and **BF**, where **BF** is the Barcan schema $\forall y \Box\phi \rightarrow \Box\forall y\phi$ of Subsection 2.9.

We briefly review Kripke semantics (with increasing and with constant domains). A Kripke *frame* $\mathfrak{F} = (W, R)$ is a graph, that is a set W endowed with a binary relation R. A

[1]Languages containing identity, functions and constant symbols will be considered only from Section 9 on.

Kripke $\mathfrak{F}$-domain (or simply a Kripke domain) D based on the Kripke frame $\mathfrak{F} = (W, R)$ is a collection of (non empty) sets $D = \{D_w \mid w \in W\}$ such that $D_v \subseteq D_w$ holds whenever vRw (i.e., whenever w is 'accessible' from v). The Kripke domain D is *constant* iff we have $D_v = D_w$ for all $v, w \in W$; for a constant Kripke domain D, the indexes $v, w, \ldots$ in $D_v, D_w, \ldots$ are usually suppressed. A Kripke *skeleton* is a pair $(\mathfrak{F}, D)$ (written also as (W, R, D)), where $\mathfrak{F} = (W, R)$ is a frame and D is a Kripke $\mathfrak{F}$-domain.

A *Kripke model* $\mathfrak{M} = (\mathfrak{F}, D, \mathcal{I})$ is a triple given by a Kripke frame $\mathfrak{F} = (W, R)$, a Kripke $\mathfrak{F}$-domain D and an *interpretation* (or valuation) function $\mathcal{I}$ mapping every n-ary predicate symbol P to a collection of subsets $\mathcal{I}(P) = \{\mathcal{I}(P)_w \subseteq D_w^n \mid w \in W\}$. For $n = 0$, by D^0 we mean a singleton Kripke $\mathfrak{F}$-domain (that is, D_w^0 contains just the empty tuple of elements from D_w); we assume that in a Kripke model, $\mathcal{I}(\bot)$ is always the collection of the empty subsets $\{\emptyset \subseteq D_w^0 \mid w \in W\}$. The Kripke model $\mathfrak{M} = (\mathfrak{F}, D, \mathcal{I})$ is said to be a *constant domain model* iff D is constant as a Kripke domain.

Given a Kripke model $\mathfrak{M} = (\mathfrak{F}, D, \mathcal{I})$, a world $w \in W$, a w-assignment g (that is a map from the set of variables to D_w) and a formula ϕ, the forcing relation $\mathfrak{M}, g, w \models \phi$ (written from now on as $g \models_w^{\mathfrak{M}} \phi$) is defined as in in Part I, Subsection 2.4. We say that ϕ is valid in $\mathfrak{M}, w$ (written $\models_w^{\mathfrak{M}} \phi$) iff $g \models_w^{\mathfrak{M}} \phi$ holds for all g and we say that ϕ is valid in $\mathfrak{M}$ (written $\models^{\mathfrak{M}} \phi$) iff $\models_w^{\mathfrak{M}} \phi$ holds for all w. We also use the notation $\models_w^{\mathfrak{M}} \phi(a_1, \ldots, a_n)$ to mean $g \models_w^{\mathfrak{M}} \phi(x_1, \ldots, x_n)$, where g is any assignment such that $g(x_1) = a_1, \ldots, g(x_n) = a_n$.

A formula ϕ is *valid in a Kripke skeleton* $(\mathfrak{F}, D)$ iff it is valid in any Kripke model of the kind $\mathfrak{M} = (\mathfrak{F}, D, \mathcal{I})$. For instance, for a given $(\mathfrak{F}, D)$, we have that **BF** is valid in $(\mathfrak{F}, D)$ iff D is constant as a Kripke domain. A modal system **S** is valid in $(\mathfrak{F}, D)$ iff all $\phi \in$ **S** are valid in $(\mathfrak{F}, D)$. The set of formulas valid in a Kripke skeleton $(\mathfrak{F}, D)$ is actually a modal system that is denoted $\mathbf{S}(\mathfrak{F}, D)$. We finally recall from Part I that **QK** (resp. **BF**.**K**) coincides with the set of formulas which are valid in all Kripke skeletons (resp. in all constant domain Kripke skeletons).

7 DECISION PROBLEMS

Although classical first-order logic is known to be undecidable, there are interesting fragments which are actually decidable: among them, we have the fragment containing only monadic predicate symbols [81], the fragment with two individual variables [107], [88], [109], the guarded and the packed guarded fragments [1], [60], [86]. The border between decidability and undecidability is, however, quite subtle: three variables and binary predicate letters are sufficient, for instance, to cross the border and consequently to get an undecidable fragment [124] (a great deal of information on the subject can be found in the monograph [15] and in [29]).

As the notion of 'one-more dimension' is implicit in modal formalisms, there is no surprise in the fact that full modal extensions of classically decidable fragments are usually no longer decidable. We first give an account of such negative results, but in the second part of this section we also investigate remarkable positive results coming from recent literature.

Recall that we already fixed a modal language $\mathcal{L}$ containing infinitely many predicate symbols of any arity, but no identity nor function or constant symbols. A *fragment* of $\mathcal{L}$ is a set F of $\mathcal{L}$-formulas; for instance, the *classical fragment* $\mathcal{L}_C$ of $\mathcal{L}$ consists of the $\Box$-free $\mathcal{L}$-formulas. If a modal system **S** and a fragment F are given, by the F-fragment

of $\mathbf{S}$ we mean the set $\mathbf{S} \cap F$ of formulas which belong to both $\mathbf{S}$ and F. Notice that this definition leaves aside interesting questions concerning the axiomatizability of $\mathbf{S} \cap F$ in terms of axiom schemata and inference rules operating only on formulas belonging to F.

A Kripke skeleton (W, R, D) is said to be *countably large* iff (i) D is constant and countable; (ii) for some $w \in W$, the set of possible worlds $R(w) := \{v \mid wRv\}$ is infinite.

The first fragment we consider is the *monadic fragment* F_m formed by the formulas of $\mathcal{L}$ containing only unary predicate symbols. The following negative result is due to Kripke [73]:

THEOREM 23. *Let $\Sigma \subseteq F_m$ be a set of sentences such that:*

(i) Σ *contains all formulas ψ such that $\psi \in F_m$ and ψ is a substitution instance of a classically valid formula of $\mathcal{L}_C$;*

(ii) $\Sigma \subseteq \mathbf{S}(W, R, D)$ *for a countably large Kripke skeleton* (W, R, D).

Then Σ is undecidable.

Proof. A formula $\psi \in \mathcal{L}_C$ is said to be dyadic iff it contains only the binary relation symbol S. Classically valid dyadic formulas form an undecidable class, so our strategy consists of reducing the decision problem for classically valid dyadic formulas to the decision problem for Σ. For dyadic ψ, let ψ^t be the formula obtained from ψ by replacing the atomic subformulas $S(x, y)$ by $\Diamond(P(x) \wedge Q(y))$, where P, Q are distinct unary predicate letters (clearly $\psi^t \in F_m$). We show that a dyadic sentence ϕ is classically valid iff $\phi^t \in \Sigma$. One direction is just by the assumption (i); for the other side, suppose that ϕ is not classically valid. By standard model theory, there is a countable classical first-order structure $\mathcal{S} = (D, I)$ such that $\mathcal{S} \not\models \phi$ (here D is the countable domain of $\mathcal{S}$ and I is the related interpretation function). Let $w \in W$ be such that $R(w)$ is infinite and let $\rho : R(w) \longrightarrow D$ be any surjection. We define a Kripke model $\mathfrak{M} = (W, R, D, \mathcal{I})$ in the following way: for $v \notin R(w)$, we let $\mathcal{I}(P)_v := \mathcal{I}(Q)_v := \emptyset$ and for $v \in R(w)$, we let $\mathcal{I}(P)_v := \{\rho(v)\}$ and $\mathcal{I}_v(Q) = \{b \in D \mid \mathcal{S} \models S(\rho(v), b)\}$. Now it is sufficient to show by induction that for every dyadic formula $\psi(\underline{x})$ and $\underline{a} \in D$, we have $\models_w^{\mathfrak{M}} \psi^t(\underline{a})$ iff $\mathcal{S} \models \psi(\underline{a})$ (in particular, ϕ^t fails in $\mathfrak{M}$ and hence cannot belong to Σ by assumption (ii)). The proof is easy: for the atomic formulas case, we have $\models_w^{\mathfrak{M}} \Diamond(P(a_1) \wedge Q(a_2))$ iff there is $v \in R(w)$ s.t. $\models_v^{\mathfrak{M}} P(a_1) \wedge Q(a_2)$ iff there is $v \in R(w)$ s.t. $\rho(v) = a_1$ & $\mathcal{S} \models S(\rho(v), a_2)$ iff there is $v \in R(w)$ s.t. $\rho(v) = a_1$ & $\mathcal{S} \models S(a_1, a_2)$ iff (ρ being surjective) $\mathcal{S} \models S(a_1, a_2)$. ❑

In particular, Theorem 23 means that any subsystem of **QS5** has an undecidable monadic fragment. The second fragment we consider is the *two-variable fragment* F_2: this is formed by the formulas of $\mathcal{L}$ containing at most two (bound or free) variables. The following negative result was obtained quite recently [71] (it generalizes previous results in [46] for systems with the Barcan formula):

THEOREM 24. *Let $\Sigma \subseteq F_2$ be a set of sentences such that*

$$\mathbf{QK} \cap F_2 \ \subseteq \ \Sigma \ \subseteq \ \mathbf{S}(W, R, D),$$

for a countably large Kripke skeleton (W, R, D). Then Σ is undecidable.

Proof. We sketch the argument of [71], which uses a reduction of an undecidable tiling problem [8]. The problem is the following: we are given a finite set T, whose members

$t = \langle u(t), d(t), r(t), l(t)\rangle$ are 4-tuple of 'colours', and we are asked about the existence of a $\mathbb{N} \times \mathbb{N}$-tiling function, i.e., of a function $\tau : \mathbb{N} \times \mathbb{N} \longrightarrow T$ such that for all $i, j \in \mathbb{N}$ we have:

$$u(\tau(i,j)) = d(\tau(i,j+1)) \quad \text{and} \quad r(\tau(i,j) = l(\tau(i+1,j)).$$

Given a finite T, let χ_T be the F_2-sentence obtained as a conjunction of (1)-(8) below:

$$\begin{array}{lr}
\forall x \bigvee_{t\in T}(P_t(x) \wedge \bigwedge_{t'\neq t} \neg P_{t'}(x)), & (1)\\
\forall x\, \forall y\, (succ_H(x,y) \to \bigwedge_{r(t)\neq l(t')} \neg(P_t(x) \wedge P_{t'}(y))), & (2)\\
\forall x\, \forall y\, (succ_V(x,y) \to \bigwedge_{u(t)\neq d(t')} \neg(P_t(x) \wedge P_{t'}(y))), & (3)\\
\forall x\, \exists y\, succ_H(x,y) \wedge \forall x\, \exists y\, succ_V(x,y), & (4)\\
\forall x\, \forall y\, (succ_H(x,y) \to \Box succ_H(x,y)) \;\wedge\; \forall x\, \forall y\, (succ_V(x,y) \to \Box succ_V(x,y)), & (5)\\
\forall x\, \forall y\, (\Diamond succ_V(x,y) \to succ_V(x,y)), & (6)\\
\forall x\, \Diamond Q(x), & (7)\\
\Box \forall x\, \forall y\, [succ_V(x,y) \wedge \exists x(Q(x) \wedge succ_H(y,x)) \to & \\
\qquad \forall y(succ_H(x,y) \to \forall x(Q(x) \to succ_V(y,x)))]. & (8)
\end{array}$$

We show that there is a tiling function $\tau : \mathbb{N} \times \mathbb{N} \longrightarrow T$ if and only if $\neg\chi_T \notin \Sigma$.

Suppose in fact that there is such a tiling τ; let $w \in W$ be a world such that $R(w)$ is infinite and let $\rho : R(w) \longrightarrow \mathbb{N} \times \mathbb{N}$ be any surjective function. Since D is countable, we can identify it with $\mathbb{N} \times \mathbb{N}$ and define a Kripke model $\mathfrak{M} = (W, R, D, \mathcal{I})$ as follows:

- $\mathcal{I}(Q)_v = \{\rho(v)\}$ if $v \in R(w)$ and $\mathcal{I}(Q)_v = \emptyset$ if $v \notin R(w)$;
- $\mathcal{I}(P_t)_v = \{\langle i,j\rangle \mid \tau(i,j) = t\}$;
- $\mathcal{I}(succ_H)_v = \{\langle (i_1,j),(i_2,j)\rangle \mid i_2 = i_1 + 1\}$ and $\mathcal{I}(succ_V)_v = \{\langle (i,j_1),(i,j_2)\rangle \mid j_2 = j_1 + 1\}$.

Since it is easy to check that $\models_w^{\mathfrak{M}} \chi_T$, we have that $\neg\chi_T \notin \mathbf{S}(W,R,D)$, i.e., $\neg\chi_T \notin \Sigma$.

Conversely, suppose that $\neg\chi_T \notin \Sigma$: this means that $\neg\chi_T \notin \mathbf{QK}$, i.e., that χ_T is satisfiable in a world v of a certain Kripke model $\mathfrak{M}$. Now the conjunction of the two-variable formulas (4)-(8) is easily seen to imply in $\mathbf{QK}$ the (three-variable) formula

$$(9) \quad \forall x\, \forall y\, \forall z\, [succ_H(x,y) \wedge succ_V(x,z) \to \exists x\, (succ_H(z,x) \wedge succ_V(y,x))].$$

Thus (9) is also true at v in the model $\mathfrak{M}$. The truth of (9) and of (4) implies that for every $i, j \in \mathbb{N}$ there are individuals a_{ij} living in v, such that $\models_v^{\mathfrak{M}} succ_H(a_{ij}, a_{i+1,j})$ and $\models_v^{\mathfrak{M}} succ_V(a_{ij}, a_{i,j+1})$. From the fact that formulas (1)-(3) hold in v, it is easily seen that the function defined by

$$\tau(i,j) = t \quad \text{iff} \quad \models_v^{\mathfrak{M}} P_t(a_{ij})$$

tiles $\mathbb{N} \times \mathbb{N}$. ❑

Since one-variable fragments of standard quantified modal systems are usually decidable [110], one may suspect that the source of undecidability is the application of modal operators to formulas with two free variables. A modal formula ϕ is said to be *monodic* iff all subformulas of ϕ whose main connective is $\Box$ contain at most one free variable. Roughly speaking, we shall prove that the decision problem for validity in the fragment

F_{mon} formed by the monodic formulas can be reduced to the decision problem for validity in the classical fragment $\mathcal{L}_C$; the proof of such a result will entail that validity in fragments like $F_m \cap F_{mon}$, $F_2 \cap F_{mon}$ is decidable. More generally, classically decidable well-behaved fragments do not loose decidability if they are extended to the modal monodic fragment (for a recent precise formulation of this fact in a general combination context, see [57]).

For simplicity, we restrict our attention to the monodic fragment over **BF**.**QK**, that is we deal with the problem of deciding whether a monodic formula is valid in all Kripke models with constant domains. It should be noticed that there are various unnecessary restrictions in this approach: (a) results can be extended to models with increasing domains and to standard systems based on $\mathbf{T}, \mathbf{S4}, \mathbf{K4}, \ldots$ (instead of $\mathbf{K}$); (b) the addition of rigid designator constants is harmless; (c) more complicated modal operators (like reflexive-transitive closures in **PDL**-style, binary temporal operators like 'since' and 'until' for some standard temporal flows semantics) can be considered, without affecting the results to be illustrated in this subsection. In some cases (like (a)-(b)), rather trivial modifications in the proofs below are required, in some other cases (like those in (c)) the extension is not straightforward at all, however the method is basically the same. The interested reader is referred to [47] for an almost complete picture of the existing results.

Let us fix for the remaining part of this subsection a monodic sentence ϕ. Let $sub(\phi)$ be the set of the subformulas of ϕ together with their negations: this set is finite, provided we 'define' $\neg\neg\psi$ to be ψ. By $sub_z(\phi)$ (where z is a variable not occurring in ϕ) we mean $\{\psi(z/x) \mid \psi(x) \in sub(\phi)$, for some variable $x\}$.[2]

For any subformula $\Box\chi$ of ϕ containing at most one free variable, let $P_{\Box\chi}$ be a predicate symbol not occurring in ϕ: the arity of $P_{\Box\chi}$ is 1 if χ contains a free variable, it is 0 (i.e., $P_{\Box\chi}$ is a propositional letter) otherwise. $P_{\Box\chi}$ is called the *surrogate* of $\Box\chi$. For any subformula ψ of ϕ, let $\overline{\psi}$ be the result of replacing the outermost subformulas of ψ whose main connective is $\Box$, by their surrogates. Clearly $\overline{\psi} \in \mathcal{L}_C$.

DEFINITION 25. A *1-type* t is any subset of $sub_z(\phi)$ such that $\{\overline{\psi} \mid \psi \in t\}$ is maximal consistent. A *world-candidate* for ϕ is any non empty set $\mathcal{T}$ of 1-types; a world-candidate is *realizable* iff the formula

$$(\alpha_{\mathcal{T}}) \qquad \bigwedge_{t\in\mathcal{T}} \exists z(\bigwedge_{\psi(z)\in t} \overline{\psi}(z)) \wedge \forall z(\bigvee_{t\in\mathcal{T}} \bigwedge_{\psi(x)\in t} \overline{\psi}(z))$$

is classically consistent.

Notice that all types belonging to the same realizable world candidate must contain the same sentences (i.e., the same subformulas of ϕ not containing free variables), by the maximality request on the definition of a 1-type. Any world w in a Kripke model $\mathfrak{M} = (\mathfrak{F}, D, \mathcal{I})$ gives rise to a realizable world-candidate, by taking the set of 1-types that are realized in it (where t is said to be realized in w iff there is $a \in D$ such that $\models_w^{\mathfrak{M}} \bigwedge_{\psi(z)\in t} \psi(a)$).

DEFINITION 26. Let $\mathfrak{F} = (W, R)$ be a Kripke frame. A *domain-candidate* is a function δ associating with every $w \in W$ a realizable world-candidate δ_w. A *run* in a domain-candidate δ, is a map r associating with every $w \in W$ a 1-type $r(w) \in \delta_w$, satisfying the

[2]Recall that we use the notation $\psi(x)$ to express the fact that in ψ *at most* the variable x is free (thus the notation $\psi(x)$ does not prevent ψ from being a sentence).

following condition for every $\Box\psi \in sub_z(\phi)$:

$$\Box\psi \in r(w) \quad \text{iff} \quad \forall v\,(wRv \;\Rightarrow\; \psi \in r(v)).$$

Finally, a *quasi-model* is a domain-candidate δ such that for every $w \in W$ and $t \in \delta_w$ there is a run r such that $r(w) = t$.

We say that ϕ is satisfied in a quasi-model (W, R, δ) iff for some $w \in W$ and $t \in \delta_w$, we have that $\phi \in t$. This notion turns out to coincide with standard satisfiability in Kripke models:

PROPOSITION 27. *ϕ is satisfied in a constant domain Kripke model $\mathfrak{M} = (W, R, D, \mathcal{I})$ iff it is satisfied in a quasi-model (W, R, δ), based on the same Kripke frame (W, R).*

Proof. One direction is trivial: if ϕ is satisfied in $\mathfrak{M} = (W, R, D, \mathcal{I})$, then it is satisfied in the quasi-model (W, R, δ), where δ_w is the function associating with $w \in W$ the set of 1-types that are realized in w.

Conversely, suppose that ϕ is satisfied in (W, R, δ). For every $w \in W$, let $\mathcal{S}_w$ be a classical first-order structure which is a model of the formula $(\alpha_{\delta(w)})$ from Definition 25; by standard classical model theory (since we are considering languages without identity), we can raise the cardinality of the support of $\mathcal{S}_w$ to any chosen infinite cardinal. Better, we may freely suppose that the support of all the $\mathcal{S}_w$ is constantly equal to $D := \kappa \times \mathcal{R}$, where $\mathcal{R}$ is the set of all runs in (W, R, δ) and κ is any infinite cardinal bigger than the cardinality of $\mathcal{R}$. Notice that any $a \in D$ is in this way a pair (a^r, a^c), where a^r is a run in (W, R, δ) and $a^c < \kappa$. Also, we can raise in $\mathcal{S}_w$ the cardinality of the set of elements satisfying each 1-type of δ_w to κ and freely suppose that, for $a \in D$, we have

(1) $\quad \mathcal{S}_w \models \overline{\psi}(a) \quad \text{iff} \quad \psi(z) \in a^r(w)$

for all $\psi(z) \in sub_z(\phi)$.

Now we can define the desired Kripke model $\mathfrak{M} = (W, R, D, \mathcal{I})$, by taking, for every n-ary predicate letter, $\mathcal{I}_w(P) := \{(a_1, \ldots, a_n) \mid \mathcal{S}_w \models P(a_1, \ldots, a_n)\}$. It is now sufficient to show, by induction, that for every subformula $\psi(x_1, \ldots, x_n)$ of ϕ and for every $a_1, \ldots, a_n \in D$, we have

(2) $\quad \models^{\mathfrak{M}}_w \psi(a_1, \ldots, a_n) \quad \text{iff} \quad \mathcal{S}_w \models \overline{\psi}(a_1, \ldots, a_n).$

We just show the $\Box$-case: since ϕ is monodic, we can suppose $n = 1$.[3] Hence we have:

$$\begin{aligned} &\mathcal{S}_w \models \overline{\Box\psi}(a_1) \;\Leftrightarrow\; \Box\psi(z) \in a_1^r(w) \Leftrightarrow \forall v(wRv \Rightarrow \psi(z) \in a_1^r(v)) \;\Leftrightarrow \\ \Leftrightarrow\;& \forall v(wRv \Rightarrow \mathcal{S}_v \models \overline{\psi}(a_1)) \;\Leftrightarrow \forall v(wRv \Rightarrow \models^{\mathfrak{M}}_v \psi(a_1)) \;\Leftrightarrow\; \models^{\mathfrak{M}}_w \Box\psi(a_1). \end{aligned}$$

(1) and (2) show, in particular, that ϕ is satisfiable in $\mathfrak{M}$, being satisfiable in (W, R, δ). ❑

The next step is to represent quasi-models satisfying ϕ as structures repeating a finite set of special patterns, called blocks. We denote by $\mathfrak{F}_n = (W_n, R_n)$ the finite Kripke frame given by $W_n := \{0, 1, \ldots, n\}$, $R_n := \{(0, i) \mid 1 \leq i \leq n\}$ (this is a finite rooted tree containing only the root 0 and the leaves $1, \ldots, n$).

[3]For $n = 0$, take an arbitrary $a_1 \in D$ (if ψ is a sentence, both relations $\models^{\mathfrak{M}}_w \psi(a_1)$ and $\mathcal{S}_w \models \overline{\psi}(a_1)$ are not influenced by the choice of a_1).

DEFINITION 28. Let $(\mathfrak{F}_n, \delta)$ be a domain candidate based on $\mathfrak{F}_n$. A *root-pseudo-run* in it is a map r associating with every $w \in W_n$ a 1-type $r(w)$ satisfying the following condition for every $\Box\psi \in sub_z(\phi)$:

$$\Box\psi \in r(0) \quad \text{iff} \quad \forall v \in W_n \ (0Rv \Rightarrow \psi \in r(v)).$$

We say that $(\mathfrak{F}_n, \delta)$ is a *block* iff for every $w \in W_n$ and $t \in \delta_w$ there is a root-pseudo-run r such that $r(w) = t$.

Again, we say that ϕ is satisfied in a block (W_n, R_n, δ) iff for some $w \in W_n$ and $t \in \delta_w$, we have that $\phi \in t$.

DEFINITION 29. A set $\mathcal{B}$ of blocks is a *satisfying set* for ϕ iff $\mathcal{B}$ contains a satisfying block for ϕ and moreover for every world w in a block $(\mathfrak{F}_n, \delta)$ belonging to $\mathcal{B}$, there is a block $(\mathfrak{F}_{n'}, \delta')$, again in $\mathcal{B}$, such that $\delta(w) = \delta'(0)$.

THEOREM 30. *ϕ is satisfiable in a Kripke model (with constant domains) iff there a satisfying set for ϕ whose blocks contain at most $2k \cdot 2^k$ worlds, where k is the cardinality of $sub_z(\phi)$.*

Proof. If there is a satisfying set for ϕ, it is not difficult to 'glue' together the various blocks in it, thus forming a satisfying quasi-model for ϕ based on an intransitive tree:[4] then Proposition 27 applies.

Conversely, if ϕ is satisfied in a Kripke model with constant domains, then (by standard modal techniques, like unravelling, or directly by the subordination frame technique of [67]) ϕ is satisfied in a model based on an intransitive tree $\mathfrak{F} = (W, R)$. By Proposition 27, ϕ is satisfied in a quasi-model $(\mathfrak{F}, \delta)$ based on $\mathfrak{F}$. By 'duplicating' some worlds, if needed, we can also suppose that if wRv holds in $\mathfrak{F}$, then there is a 'twin' of v, namely a world $v' \neq v$ such that wRv' and $\delta_v = \delta_{v'}$. We now extract from $(\mathfrak{F}, \delta)$ a satisfying set for ϕ: this is done by associating with any $w \in W$ a suitable block, matching the required cardinality conditions. For every 1-type $t \in \delta_w$ and for every $\Box\psi \in sub_z(\phi)$ with $\Box\psi \notin t$, select twin worlds v_1, v_2 such that wRv_1, wRv_2 and a 1-type $t' \in \delta_{v_1} = \delta_{v_2}$ such that: (i) $\psi \notin t'$ and (ii) for all $\Box\chi \in sub_z(\phi)$, if $\Box\chi \in t$, then $\chi \in t'$ (this is possible for instance by considering any run r in $(\mathfrak{F}, \delta)$ such that $r(w) = t$ and by taking $v_1 := v$, $v_2 :=$ a twin of v_1 and $t' := r(v)$, where v is any world such that wRv and $\psi \notin r(v)$, see the definition of a run in a quasi-model).

Consider now the subframe $\mathfrak{F}_w = (W_w, R)$ of $\mathfrak{F}$ formed by w and by the worlds so selected: as $\mathfrak{F}$ is an intransitive tree, this is isomorphic to a frame of the kind $\mathfrak{F}_n$ for $n \leq 2k \cdot 2^k$. If we show that this subframe $\mathfrak{F}_w$ is a block $\mathcal{B}_w$ (with respect to the restriction of δ), the theorem is proved (the required satisfying set for ϕ is formed by the various blocks $\mathcal{B}_w$, varying $w \in W$). Take a type $t \in \delta_v$ for a possible world v in $\mathcal{B}_w = (W_w, R, \delta)$; let r be a run in $(\mathfrak{F}, \delta)$ such that $r(v) = t$. Consider now the 1-type $r(w)$: by the above construction of W_w, for each $\Box\psi \in sub_z(\phi)$ with $\Box\psi \notin r(w)$, there is a possible world $v_\psi \in W_w$ with wRv_ψ and a 1-type $t_\psi \in \delta_{v_\psi}$ such that: (i) $\psi \notin t_\psi$ and (ii) for all $\Box\chi \in sub_z(\phi)$, if $\Box\chi \in r(w)$, then $\chi \in t_\psi$. As in fact there are such twin v_ψs to choose between, we can assume that $v_\psi \neq v$ for every ψ. Define now the root-pseudo-run s by letting $s(u) := t_\psi$ if $u = v_\psi$ for some ψ, $s(u) := r(u)$, otherwise. Since v is not among the v_ψ's, this implies $s(v) = r(v) = t$, as required. ❑

[4]A Kripke frame (W, R) is said to be an intransitive tree iff there is a world $w_0 \in W$ such that for every $v \in W$, there is exactly one path $w_0 R w_1 R \cdots R w_n = v$ ($n \geq 0$) from w_0 to v.

COROLLARY 31. *Let $F \subseteq F_{mon}$ be a subfragment of the monodic fragment; suppose that for a sentence $\phi \in F$ there is an algorithm that decides whether a world-candidate for ϕ is realizable or not. Then the set of sentences in $F \cap$ **BF.K** is decidable.*

Thus, in particular, $F_m \cap F_{mon} \cap$ **BF.K** and $F_2 \cap F_{mon} \cap$ **BF.K** are decidable, because for $\phi \in F_m \cap F_{mon}$ (resp. $\phi \in F_2 \cap F_{mon}$), the formulas of the kind $(\alpha_{\mathcal{T}})$ are still in $F_m \cap F_{mon}$ (resp. in $F_2 \cap F_{mon}$). Corollary 31 applies also to the monodic guarded fragment (see [47]). It should be noticed that, although the monodic fragment of **BF.K** has the finite frame property [130] (i.e., a monodic satisfiable formula can be satisfied in a Kripke model based on a finite frame), the same does not apply e.g. to **BF.S4**: for this system it is impossible to extract a finite transitive model from the construction of Theorem 30, because e.g. the formula (3) from Section 8 requires infinitely many worlds to be falsified. Notice also that, for more sophisticated systems, most (although not all) of the proofs of the incompleteness results of next section apply already to the one-variable fragment.

8 COMPLETENESS, INCOMPLETENESS AND NON-AXIOMATIZABILITY

A first-order modal system **S** is complete with respect to a class $\mathcal{K}$ of Kripke skeletons iff **S** is valid in every Kripke skeleton belonging to $\mathcal{K}$ and, moreover, every formula $\phi \notin$ **S** fails in a Kripke model $\mathfrak{M} = (\mathfrak{F}, D, \mathcal{I})$ such that $(\mathfrak{F}, D)$ belongs to $\mathcal{K}$. **S** is said to be *K(ripke)-complete* iff it is complete with respect to some class $\mathcal{K}$ of Kripke skeletons.

Basic quantified systems like **QK**, **QT**, **QK4**, **QS4**, **QS5**, etc., as well as their variants with the Barcan formula, are all Kripke complete (see [25], [68] or Chapter 2 of this Handbook). However, despite the simplicity of the definition of a K-complete modal system, we shall show that K-completeness (contrary to the propositional case) is not as frequent as one might expect.

We first mention a couple of examples of K-complete systems, whose completeness proofs are not immediate (and which were open problems till the late eighties). Recall that **S4** is obtained from **K** by adding it the axiom schemata $\Box\phi \to \phi$, $\Box\phi \to \Box\Box\phi$; from the semantic side, **S4** is valid and complete with respect to Kripke frames $(W, \leq)$ in which the accessibility relation $\leq$ is a *preorder*, i.e., it is reflexive and transitive (when the letter $\leq$ is used for the accessibility relation of a frame, it is implicitly assumed that such an accessibility relation is a preorder). Recall also that **S4.2** is the extension of **S4** axiomatized by the schema $\Diamond\Box\phi \to \Box\Diamond\phi$ and that **S4.3** is the extension of **S4** axiomatized by the schema $\Box(\Box\phi \to \psi) \vee \Box(\Box\psi \to \phi)$.[5]

THEOREM 32. *The systems* **QS4.2** *and* **QS4.3** *are both K-complete.*

In the case of **QS4.2** the completeness proof is obtained in [24] by adapting the subordination frame technique from [67], whereas in the case of **QS4.3**, one needs more sophisticated methods, like Fine's diagrams [22] (but see also [118]). However, the following theorem seems to suggest that K-incompleteness is the rule for many systems extending **QS4**:

THEOREM 33. [52] *Let* $\mathbf{L} \supseteq$ **S4** *be a propositional logic such that* $\mathbf{L} \not\supseteq$ **S5** *and such that* **QL** *is K-complete. Then* $\mathbf{L} \subseteq$ **S4.3** *and, if* $\mathbf{L} \neq$ **S4.3**, *then* **L** *is an unbounded width logic*

[5] See Chapter 7 for more information about the propositional systems we mention here.

(that is, for every n, there is a rooted frame[6] *for* **L** *containing n mutually incomparable states).*

The proof of Theorem 33 requires some powerful alternative semantics, like presheaf semantics (over an arbitrary category). The latter is analyzed in more detail in the topos-theoretic context of Subsection 10.1, here we use a simple formulation in terms of functional frames. Following [128], we call *functional frame* a pair $(W, \mathcal{R})$, where W is a family $\{D_w \mid w \in W\}$ of individual domains and $\mathcal{R}$ is a family of maps between such domains. Since we want **S4**-axioms to be valid,[7] we ask for $\mathcal{R}$ to contain identities and to be closed under compositions. A functional frame $(W, \mathcal{R})$ becomes a *functional model* $\mathfrak{M} = (\mathfrak{F}, D, \mathcal{I})$ upon addition of an interpretation function $\mathcal{I}$, mapping n-ary predicates P to families of subsets $\mathcal{I}_w(P) \subseteq D_w^n$. Truth of a formula ϕ in the model $\mathfrak{M}$ at a world w under the w-assignment g (in symbols $g \models_w^{\mathfrak{M}} \phi$) is defined in the expected way:[8] for instance, $g \models_w^{\mathfrak{M}} \Box\phi$ holds iff $f \circ g \models_v^{\mathfrak{M}} \phi$ holds for all $f \in \mathcal{R}$ having domain D_w and codomain D_v. Thus functional models and frames differ from the corresponding Kripke models and frames because, given two possible worlds w and v, there are many different ways of making a transition from w to v: the difference with Kripke semantics is sensible, both from a philosophical and from a mathematical point of view.

For space reasons, we cannot report here the full proof of Theorem 33, however we can illustrate the method by sketching in some detail the proof a weaker (but still informative) result. The key idea is to use an indirect kind of reduction of functional frames to ordinary Kripke frames. Let us call *frame representation* $\mathfrak{F}(W, \mathcal{R})$ of a functional frame $(W, \mathcal{R})$ the frame having the $f \in \mathcal{R}$ as states and as accessibility relation the divisibility relation (namely $f_1 \leq f_2$ holds in $\mathfrak{F}(W, \mathcal{R})$ iff we have $f_2 = h \circ f_1$, for some domain and codomain matching function $h \in \mathcal{R}$).

LEMMA 34. *Let a functional frame $(W, \mathcal{R})$ be given; if $\mathfrak{F}(W, \mathcal{R}) \models \psi$ holds for some modal propositional formula ψ, then every predicate-logical substitution instance $\sigma(\psi)$ of ψ is valid in all the functional models based on $(W, \mathcal{R})$.*

Proof. We argue by contraposition: suppose that we have $g \not\models_w^{\mathfrak{M}} \sigma(\psi)$ for some w, g in a model $\mathfrak{M}$ based on $(W, \mathcal{R})$. Define a propositional valuation V on $\mathfrak{F}(W, \mathcal{R})$ by setting (for every propositional variable p) $V(p) := \{f : D_w \longrightarrow D_v \mid f \circ g \models_v^{\mathfrak{M}} \sigma(p)\}$. By induction, it is easy to see that, for all $f \in \mathcal{R}$ having domain D_w and for all propositional modal formula χ, we have that χ is true at f in the propositional Kripke model $(\mathfrak{F}(W, \mathcal{R}), V)$ iff $f \circ g \models_v^{\mathfrak{M}} \sigma(\chi)$ holds (here D_v is the codomain of f). In particular, we get that ψ fails at id_w in $(\mathfrak{F}(W, \mathcal{R}), V)$. ❑

Let $\mathbf{L} \supseteq \mathbf{S4}$ be a propositional logic such that $\mathbf{L} \not\supseteq \mathbf{S5}$ and such that **QL** is K-complete: *we shall prove that* $\mathbf{L} \subseteq \mathbf{S4.3.Grz}$ (equivalently, that every finite chain is a frame for **L**). Bearing this aim in mind, consider the following formula ϕ

(3) $\Box\forall x(P(x) \to \Box P(x)) \to \Diamond\forall x(\Diamond P(x) \to P(x))$.

Now, if $\mathbf{QL} \not\vdash \phi$, by the Kripke completeness for **QL**, it is clear that there exists a Kripke skeleton $(\mathfrak{F}, D)$ (based on a Kripke frame $\mathfrak{F}$ for **L**) containing a generated subframe

[6] A preordered set $(W, \leq)$ is rooted iff for some $\rho \in W$, we have $\rho \leq v$ for all v.

[7] This is a simplified analysis, in fact correspondence theory for functional frames still needs full investigation (for some subtleties arising here, see [128] again).

[8] In particular, quantification at w ranges over the domain D_w only.

that can be p-morphically mapped onto any finite linear chain,[9] which means that $\mathbf{L} \subseteq \mathbf{S4.3.Grz}$, as claimed. Thus, it is sufficient to show that $\mathbf{QL} \vdash \phi$ implies $\mathbf{L} \supseteq \mathbf{S5}$: here functional models comes into the picture.

Consider the following functional frame $(W, \mathcal{R})$: W contains just one world w, the unique individual domain is $D_w := \{1, 2\}$ and the functions in $\mathcal{R}$ are the identity function and the constant function with value 2. The formula ϕ is not valid in a functional model $\mathfrak{M}$ based on $(W, \mathcal{R})$ (take $\mathcal{I}_w(P) := \{2\}$); since we supposed $\mathbf{QL} \vdash \phi$ and functional models semantics is obviously valid, some predicate-logical substitution instance of an axiom of $\mathbf{L}$ fails in $\mathfrak{M}$. By Lemma 34, we must conclude that $\mathfrak{F}(W, \mathcal{R})$ is not a frame for $\mathbf{L}$. However, $\mathfrak{F}(W, \mathcal{R})$ is the two-element chain and consequently $\mathbf{L} \supseteq \mathbf{S5}$, as claimed.

The full statement of Theorem 33 is obtained by repeating again and again the above schema (find a formula that functional models prove to be independent through Lemma 34 and identify some necessary condition for it to be false in a Kripke skeleton). Additional incompleteness results can be found in [111]; we mention just one of them: the system axiomatized by adding to $\mathbf{QS4}$ the modal translation of the 'constant domain' superintuitionistic axiom schema $\forall y\, (\psi(y, \underline{x}) \vee \phi(\underline{x})) \to (\forall y\, \psi(y, \underline{x})) \vee \phi(\underline{x})$ is unable to prove the Barcan formula and hence it is not K-complete.

For constant domain semantics, a general completeness/incompleteness theorem is available, by combining the results from [125] and [113]:

THEOREM 35. *If* $\mathbf{L} \supseteq \mathbf{S4}$ *is a subframe logic, then* $\mathbf{BF.L}$ *is K-complete iff* $\mathbf{L}$ *has the finite embedding property.*[10]

However, it seems to be problematic to get positive results beyond the subframe logics case: the system $\mathbf{BF.S4.2}$ is not K-complete [111], [50] and $\mathbf{BF.S4.1}$ (namely $\mathbf{BF.S4}$ plus $\Box\Diamond\phi \to \Diamond\Box\phi$) is incomplete as well [68].

K-completeness can be re-gained by axiomatizing Kripke completions: the *Kripke completion* of a modal system $\mathbf{S}$ is the modal system $\mathbf{S_K}$ containing the formulas which are valid in all the Kripke skeletons in which $\mathbf{S}$ is valid. Some work has been done for the axiomatization of Kripke completions within the related field of Kripke semantics for *intermediate* predicate logics: for instance, an axiomatization of the intuitionistic first-order formulas which are valid in Kripke models based on posets not exceeding a preassigned bounded height is given in [131], where the insufficient propositional axioms are suitably strengthened. If one restricts to the intuitionistic first-order formulas which are valid in Kripke models based on a given *single finite poset*, then general reasons (which apply to any first-order axiomatizable class of Kripke skeletons) guarantee recursive enumerability, whereas nothing can be said about existence of a finite axiomatization: the latter exists for the constant domain case (where it can be shown that propositional axioms suffice [112]), but not always for the nested domain case, see [95], [115], [116], [117] for existing results on the subject.

The problem of axiomatizing an (even quite natural) given class of Kripke skeletons might be a tremendous task, as exemplified by the case of constant domain skeletons based on $(\mathbb{N}, \leq)$:[11]

[9]Starting from any world w falsifying ϕ, we can in fact find an infinite strictly ascending chain of worlds over w, thus p-morphisms from the w-upper cone onto finite linear chains easily obtain.

[10]See Chapter 7 for the definition of a subframe logic. $\mathbf{L}$ is said to have the finite embedding property iff every frame in which $\mathbf{L}$ is not valid contains a finite subframe in which $\mathbf{L}$ is not valid as well.

[11]Theorem 36 comes from unpublished work by D. Scott. Scott's method is outlined in [45] (here we give a similar, but simplified argument).

THEOREM 36. *The set of modal formulas which are valid in constant domain models based on the frame* $(\mathbb{N}, \leq)$ *is not arithmetical (hence, a fortiori, not recursively enumerable).*

Proof. We reduce true arithmetic to validity in constant domain models based on $(\mathbb{N}, \leq)$. For the purposes of this proof only, let us assume for simplicity that our language $\mathcal{L}$ contains equality, constant and function symbols[12] (these are treated rigidly in models, contrary to the extensions of predicates which are allowed to vary). Let $\mathcal{L}_R$ be the sublanguage of $\mathcal{L}$ containing equality, a constant $\overline{0}$, a unary function symbol s and binary function symbols $+, \cdot$; $\mathcal{L}_R$ can be used in order to introduce Robinson's arithmetic (see e.g. [90]): this is the first-order theory obtained from Peano's arithmetic by replacing the induction schema by the formula $\forall x(x = \overline{0} \vee \exists y(x = s(y)))$. Robinson arithmetic is a finitely axiomatized theory (let ρ be the conjunction of the related axioms), in which recursive functions and predicates are formally representable. In particular, for all $n, m \in \mathbb{N}$, we have (we use $\vdash$ for provability in classical logic):

$$n \neq m \quad \Rightarrow \quad \vdash \rho \to \overline{n} \neq \overline{m}; \tag{4}$$

$$\vdash \rho \to \overline{n} + \overline{m} = \overline{n+m}; \tag{5}$$

$$\vdash \rho \to \overline{n} \cdot \overline{m} = \overline{n \cdot m}. \tag{6}$$

Moreover, if we define $x \leq y$ as $\exists z(z + x = y)$, it is well-known (and easily proved by metatheoretical induction over m) that we have:

(7) $\vdash \rho \to \forall x\,(x \leq \overline{m} \leftrightarrow (x = \overline{0} \vee \cdots \vee x = \overline{m}))$.

Let N be a unary predicate (not in $\mathcal{L}_R$) and let τ be the conjunction of the following three formulas

$$\forall x \forall y(y \leq x \leftrightarrow \Box(N(x) \to N(y))), \quad \forall x \Box(N(x) \to \Box N(x)), \quad \forall x \Diamond N(x).$$

For every ($\Box$-free) sentence ϕ in $\mathcal{L}_R$, we show that ϕ is true in the standard model $\mathbb{N}$ of arithmetic iff $\rho \wedge \tau \wedge \phi$ is satisfiable in a constant domain model based on $(\mathbb{N}, \leq)$ (hence, $\neg\phi$ is a true arithmetical statement iff $\neg(\rho \wedge \tau \wedge \phi)$ is valid in our semantics). One direction is easy: if ϕ is a true arithmetical sentence, let us define $D := \mathbb{N}$ and $\mathcal{I}_n(N) := \{0, 1, \dots, n\}$ (obviously, $\mathcal{I}_n(\overline{0})$, $\mathcal{I}_n(s)$, $\mathcal{I}_n(+)$, $\mathcal{I}_n(\cdot)$ are number 0, successor, sum and product, respectively). For this $\mathfrak{M} = (\mathbb{N}, \leq, D, \mathcal{I})$, we clearly have that $\models_0^{\mathfrak{M}} \rho \wedge \tau \wedge \phi$.

Conversely, assume $\rho \wedge \tau \wedge \phi$ is satisfied in a constant domain model $\mathfrak{M} = (\mathbb{N}, \leq, D, \mathcal{I})$ (without loss of generality, we may suppose that $\models_0^{\mathfrak{M}} \rho \wedge \tau \wedge \phi$). Then $(D, \mathcal{I}_0)$ is a classical first-order structure which is a model of Robinson arithmetic. Because of that, the map associating $\mathcal{I}_0(\overline{n})$ with n is an injective homomorphism of the standard model $\mathbb{N}$ into $(D, \mathcal{I}_0)$ (see (4)-(6)). Thus, to show that ϕ is indeed a true arithmetical sentence, we only need to check that D does not contain non-standard elements, i.e., that for any $b \in D$ there is n such that $b = \mathcal{I}_0(\overline{n})$. Here we use the fact that $\models_0^{\mathfrak{M}} \tau$ holds. Notice first that, because of this, we have $D = \bigcup_{n \in \mathbb{N}} \mathcal{I}_n(N)$ and moreover $\mathcal{I}_n(N) \subseteq \mathcal{I}_m(N)$, whenever $n \leq m$. Also, $\models_0^{\mathfrak{M}} \tau$ forces, in the world 0, the relation $x \geq y$ to be $\Box(N(x) \to N(y))$

[12] This assumption makes the arguments below more transparent, but it is not essential: one can use $n+1$-ary predicate symbols instead of n-ary function symbols, select a binary predicate symbol $\approx$ in $\mathcal{L}$ and let it play the role of equality (this means that suitable rigidity, congruence and functionality axioms for the symbols of $\mathcal{L}_R$ w.r.t. $\approx$ are included in the conjuncts of the formula ρ below).

(that is, the relation $x \geq y$ holds iff x is added to the extension of the predicate N at the same time or after y is added).

Let now k_0 be the minimum $k_0 \in \mathbb{N}$ such that $\mathcal{I}_{k_0}(N)$ is not empty: from $c \in \mathcal{I}_{k_0}(N)$, we get that c is the $\leq$-smallest element in D, hence in particular $\models_0^{\mathfrak{M}} c \leq \overline{0}$, i.e., $c = \mathcal{I}_0(\overline{0})$ by (7). Then, let $k_1 > k_0$ be the minimum number such that $\mathcal{I}_{k_1}(N)$ is a proper superset of $\mathcal{I}_{k_0}(N)$: here just $\mathcal{I}_0(\overline{1})$ is the new element entering into $\mathcal{I}_{k_1}(N)$, because from $c \in \mathcal{I}_{k_1}(N)$, we get that c is $\leq$-smaller than every element of D different from $\mathcal{I}_0(\overline{0})$, hence in particular $\models_0^{\mathfrak{M}} c \leq \overline{1}$, i.e., $c = \mathcal{I}_0(\overline{0})$ or $c = \mathcal{I}_0(\overline{1})$, by (7) again. If we proceed in this way, we identify an infinite sequence of natural numbers $k_0 < k_1 < k_2 \cdots$ such that for all n

$$\mathcal{I}_{k_n}(N) = \mathcal{I}_{k_n+1}(N) = \cdots = \mathcal{I}_{k_{n+1}-1}(N) = \{\mathcal{I}_0(\overline{0}), \ldots, \mathcal{I}_0(\overline{n})\}.$$

Let now $b \in D$: from $\models_0^{\mathfrak{M}} \Diamond N(b)$, it follows that $b = \mathcal{I}_0(\overline{n})$, for some n. ❑

Theorem 36 is just an example of a non-axiomatizable natural class of Kripke models: for non-axiomatizability of Kripke models based on finite frames, see [5], [4], [26]. Further very strong results have been reached in [119], [120], where it is shown in particular that *for any infinite family Y of finite rooted posets*, the set of valid formulas in models based on members of Y (both with constant and with nested domains) is not recursively enumerable. For non recursive enumerability of valid formulas on every noetherian frame (both with increasing and constant domains), see [105], [106].

The reader might have the impression that incompleteness and non-axiomatizability arise only in rather specific situations and that common standard semantic classes are on the contrary nicely behaved. This is not the case: so far, we examined only unary monomodal systems, but if we drop such a restriction, negative results arise even more easily. The following sample illustrates what happens to quantified extensions of **PDL**-like propositional systems. Let us consider the bimodal language $\mathcal{L}^*$ obtained from $\mathcal{L}$ by adding it an extra box operator $\Box^*$. At the semantical level, we do not need to modify the definitions of a frame, of a skeleton and of a model we gave in Section 6 for monomodal systems over **QK**, we simply add the truth-clause

$$g \models_{w_0}^{\mathfrak{M}} \Box^* \phi \qquad \text{iff} \qquad \forall n \geq 0, \forall w_1, \ldots, w_n \; (\bigwedge_{i=0}^{n-1} w_i R w_{i+1} \;\Rightarrow\; g \models_{w_n}^{\mathfrak{M}} \phi),$$

saying that $\Box^*$ is interpreted by using the reflexive-transitive closure of R. Here not only the standard propositional axioms (those listed e.g. in Chapter 12) are insufficient, but once again incompleteness cannot be repaired:

THEOREM 37. *The set of $\mathcal{L}^*$-formulas which are valid in the above semantics (both with increasing and with constant domains) is not recursively enumerable.*

The proof of this theorem is obtained in [130] by reduction of the $\mathbb{N} \times \mathbb{N}$-recurrent tiling problem. Actually, Theorem 37 applies to the two-variable monadic fragment, but *not* to the monodic fragment. The latter is nicely axiomatizable: to this end, it is sufficient [47] to join a propositionally complete set of axiom schemata with the standard quantifier laws (proofs in the resulting system are restricted to formulas belonging to the monodic fragment).

9 TOWARDS MORE POWERFUL SEMANTICS

In this section, we begin our investigation of alternative semantics, starting from an appropriate algebraic framework, namely hyperdoctrines (traditional algebraic semantics [102] is inadequate because, contrary to the propositional case, it is not complete [94]). From hyperdoctrines, we derive the most powerful existing semantics for modal first-order logic, namely metaframes; we postpone to Section 10 the analysis of suitable mathematical frameworks giving rise to further very natural extended semantics. Finally, in Section 11, we analyze counterpart semantics.

9.1 Typed Languages and Systems

Types are commonly used in higher order formalisms, where they are needed to deal with constructors like exponentiation and powerset, see e.g. [76]. The reason why we introduce them in a first order framework is different. It is well-known that a first-order formula ϕ, once interpreted in a model $\mathfrak{M}$, defines a subset $[\![\phi]\!]^{\mathfrak{M}}$ of a suitable cartesian power of the domain (of the domains, in the many-sorted case). What exactly this cartesian power is, however, is not specifically indicated in the formula itself: if ϕ contains n free variables, then $[\![\phi]\!]^{\mathfrak{M}}$ can be seen as a subset of any m-th cartesian power of the domain, where $m \geq n$. Thus, varying m, $[\![x = 0]\!]^{\mathfrak{M}}$ may denote in analytic geometry a point, a straight line, a plane, and so forth: logicians usually argue that this confusion is harmless, because infinitary assignments (not finitary assignments) are commonly used in the inductive definition of truth of a formula in a model. There is however something artificial in this choice of leaving the 'dimension' unspecified: in fact, there is an implicit notion of dimension even in contexts which are far from mathematics (for instance, dimension seems to naturally arise in linguistics, in relation to indexicals [19], see also Section 11 below). Although ordinary untyped languages are more manageable from a proof-theoretic point of view, we shall adopt typed languages from now on, because using such languages it is *easier* to interpret first-order formulas in the extended semantics that have been proposed in the literature. Typed languages will also be *indispensable* for the counterpart semantics of Section 11. We formally introduce the notion of a typed formula (i.e., of a 'formula-with-dimension').

For simplicity, we shall give the definition of a typed language in the one-sorted case only: extensions to the many-sorted case [51] are important, but straightforward and will be left to the experienced reader. Given that only one sort is allowed, types (i.e., 'dimensions') are formal cartesian products of that unique sort and can hence be identified with natural numbers greater or equal to 0.

We suppose that a first-order *language* $\mathcal{L}$ is given, where a language is now a set of *functions* and of *predicate* symbols, both endowed with a specified arity. We assume that $\mathcal{L}$ always contains a 0-ary predicate symbol $\perp$ (the 'falsehood') and a 2-ary predicate symbol $=$ (the 'equality predicate').

Terms are built up in the customary way, using countably many variables $x_1, x_2, \ldots$. A term of type n (briefly an *n-term*) is a term t in which at most the variables $x_1, \ldots, x_n$ occur; we write $t : n$ to mean that t is an n-term. If $t : n$ is an n-term and if $v_1 : m, \ldots, v_n : m$ are m-terms, $t[v_1, \ldots, v_n] : m$ is the m-term resulting from t by replacing x_i by v_i ($i = 1, \ldots, n$).

DEFINITION 38. The notion of a formula of type n (briefly an *n-formula*) $\phi : n$ is so

defined:

(i) if P is a predicate symbol of arity k and if $t_1 : n, \ldots, t_k : n$ are n-terms, then $P(t_1, \ldots, t_k) : n$ is an (atomic) n-formula;

(ii if $\phi : n$ and $\psi : n$ are n-formulas, so is $(\phi \to \psi) : n$;

(iii) if $\phi : n{+}1$ is an $n{+}1$-formula, then $(\forall x_{n+1}\phi) : n$ is an n-formula;

(iv) if $\phi : n$ is an n-formula, then $(\Box\phi) : n$ is an n-formula.

Thus quantification is allowed only with respect to the maximum index variable (this limitation does not cause any loss of expressivity, only alphabetic variants are lost). The inductive definition of an n-formula in the $\Box$-case will be modified in Section 11 in order to make it suitable for counterpart semantics. The remaining logical operators (namely $\top, \wedge, \vee, \neg, \leftrightarrow, \exists, \Diamond$) are defined in the usual way. We formally introduce *substitutions*:

DEFINITION 39. If $\phi : n$ is an n-formula and if $\underline{v} : m$ is an n-tuple of m-terms, the m-formula $\phi[\underline{v}] : m$ is so defined, by induction on $\phi : n$:

(a) if $\phi : n$ is the atomic formula $P(t_1, \ldots, t_k) : n$, then $\phi[\underline{v}] : m$ is $P(t_1[\underline{v}], \ldots, t_k[\underline{v}]) : m$;

(b) if $\phi : n$ is $\psi_1 \to \psi_2 : n$, then $\phi[\underline{v}] : m$ is $(\psi_1[\underline{v}]) \to (\psi_2[\underline{v}]) : m$;

(c) if $\phi : n$ is $\forall x_{n+1}\psi : n$, then $\phi[\underline{v}] : m$ is $\forall x_{m+1}\ (\psi[\underline{v}, x_{m+1}]) : m$;

(d) if $\phi : n$ is $\Box\psi : n$, then $\phi[\underline{v}] : m$ is $\Box(\psi[\underline{v}]) : m$.

Thus bound variables are systematically renamed in substitutions (observe also that Definition 39(c) is correct, because if the $\underline{v}$'s are m-terms, then they are also $m+1$-terms).[13] The following compatibility condition between substitutions into terms and substitutions into formulas can be easily proved by induction:

$$(\phi[t_1, \ldots, t_k])[\underline{v}] : m \ = \ \phi[t_1[\underline{v}], \ldots, t_k[\underline{v}]] : m.$$

We can emphasize the fact that a given $n{+}1$-formula does not contain occurrences of a variable, say x_{n+1}, by writing it in the form $\phi[x_1, \ldots, x_n] : n{+}1$, for a suitable n-formula $\phi : n$. Notice also that in order to take the implication of $\phi : n$ and $\psi : n{+}1$, one must e.g. use $\phi[x_1, \ldots, x_n] \to \psi : n{+}1$ (the $n+1$-formula $\phi[x_1, \ldots, x_n] : n{+}1$ differs from the n-formula $\phi : n$ because the bound variables x_i, for $i > n$, are suitably renamed). In order to make notation easier, we shall also make the following convention: by $\underline{x}$ we indicate a list of variables like $x_1, \ldots, x_n$ (for some $n \geq 0$) and, whenever the notation $\underline{x}$ is used, y stands for x_{n+1}, whereas z_1, z_2 stand for x_{n+1}, x_{n+2}, etc.

We shall make use of modal systems formulated within typed languages; from now on, we shall also consider *only systems on a propositional* **S4** *basis*. The latter choice is due to simplicity and uniformity reasons: mathematical 'topological-like' semantics require **S4**-axioms to be valid and certain arguments would look unnatural and distorted if 'topological-like' conditions would systematically be weakened in the style of neighborhood semantics for propositional modal logic. On the other hand, we shall explicitly

[13]In order to make clearer the comparison with the Beck-Chevalley property below, notice that the right-most formula $\forall x_{m+1}\ (\psi[\underline{v}, x_{m+1}]) : m$ of Definition 39(c) can also be written as $\forall x_{m+1}(\psi[\underline{v}[x_1, \ldots, x_m], x_{m+1}]) : m$, because $\underline{v}[x_1, \ldots, x_m] = \underline{v}$.

alert the reader whenever the adaptation of the semantics we introduce to systems weaker than **S4** is non trivial or problematic.

The quantified (typed) deductive system **QS4** is explained in Table 1. The instantiation rule (*Inst*) can be avoided by closing under instantiation the axiom schemata ($\forall$-*Ex*), (*Refl*), (*Repl*) (that is, one may use e.g. the schema $\forall y\phi \to \phi[\underline{x}, t] : n$ instead of ($\forall$-*Ex*)), however we prefer to keep rule (*Inst*) in order to make next subsections proofs smoother.

If we need to consider modal first-order systems **S** stronger than **QS4**, we simply add to Table 1 a set of formulas Φ *closed under uniform substitution* as extra **S**-axioms.[14] In particular, if Φ is the set of first-order instances of the formulas provable in a propositional modal logic $\mathbf{L} \supseteq \mathbf{S4}$, we get the system **QL** (this is the typed version with equality of the system **QL** of Section 6).

Typed systems and untyped ordinary systems are *equivalent formalisms*: since untyped formulas have alphabetic variants that can be typed, we have syntactic translations in both directions which reduce to identity (up to provability) in case they are sequentially applied. This result is not difficult to prove (see [51] for the details); notice however that in order to get it, the extra single axiom $\exists x_1 \top : 0$ must be added to Table 1, in case the language does not contain any constant symbol (this is because the calculus of Table 1 is sound with respect to empty domain models and consequently it cannot prove any theorem of the kind $\exists x_1 \phi : 0$ if no constant is available).

Table 1.

Axiom Schemata

$\phi : n$ (*Taut*)

(provided ϕ is an instance of a propositional **S4**-valid formula)

$(\forall y\, \phi)[\underline{x}] \to \phi : n{+}1$ ($\forall$-*Ex*)

$x_1 = x_1 : 1$ (*Refl*)

$z_1 = z_2 \to (\phi[\underline{x}, z_1] \to \phi[\underline{x}, z_2]) : n{+}2$ (*Repl*)

Inference Rules

$$\frac{\psi : n \qquad \psi \to \phi : n}{\phi : n} \quad (MP)$$

$$\frac{\phi : n}{\Box\phi : n} \quad (Nec)$$

$$\frac{\phi[\underline{x}] \to \psi : n{+}1}{\phi \to \forall y\, \psi : n} \quad (\forall\text{-}In)$$

$$\frac{\phi : n}{\phi[\underline{t}] : k} \quad (Inst)$$

(where $\underline{t}$ is an n-tuple of k-terms).

[14] The definition of uniform substitution $\phi[\psi/P]$ from Subsection 2.3 of Part I can be easily adapted to typed languages.

9.2 Cartesian and Lex Categories

We shall use hyperdoctrines in order to introduce all the alternative semantics from a unitary point of view: this will greatly simplify proofs and will be useful to explain appropriately the main phenomena arising from the interplay of modalities, substitutions, quantifiers and equality. Since we do not assume from the reader previous knowledge about basic category-theoretic concepts, we shall briefly introduce them in this subsection (more examples and motivations can be found in textbooks like [83], [14]).

A *category* $\mathbf{T}$ consists of two classes T_0 and T_1, the class of the objects and the class of the arrows of $\mathbf{T}$, respectively. To each arrow $f \in T_1$, two objects $d(f)$ and $c(f)$ are assigned (we write $f \in Hom(Y, X)$ or $f : Y \longrightarrow X$ or $Y \xrightarrow{f} X$ to mean that $d(f) = Y$ and $c(f) = X$). Moreover for each object X an arrow id_X of domain and codomain X is given and finally to each pair of arrows g, f such that $c(g) = d(f)$ an arrow $f \circ g$ of domain $d(g)$ and codomain $c(f)$ is assigned, as schematized in the following picture:

$$(Z \xrightarrow{g} Y \xrightarrow{f} X) \qquad \longmapsto \qquad Z \xrightarrow{f \circ g} X.$$

The following associativity and unity requirements must be satisfied by the above data:

$$(f \circ g) \circ h \ = \ f \circ (g \circ h), \qquad id_X \circ f \ = \ f \ = \ f \circ id_Y$$

for $W \xrightarrow{h} Z \xrightarrow{g} Y \xrightarrow{f} X$. The typical example of a category is the category **Set** of sets and functions; notice however that also a single monoid is a category (in fact monoids can be identified with categories having just one object) and that a single preordered set is a category (preordered sets can be identified with categories in which, for given objects X, Y, there is at most one arrow $f \in Hom(Y, X)$).

Objects in a category are abstract entities, hence they 'do not have elements'; however we show how the formalism of elements can be re-gained, so that it is possible up to a certain extent to work in an arbitrary category $\mathbf{T}$ as if $\mathbf{T}$ were just **Set**. For given objects U, X of $\mathbf{T}$, an U*-element* of X is, by definition, an arrow $x : U \longrightarrow X$.[15] To emphasize that $x : U \longrightarrow X$ is a 'generalized element of X', we may write $x \in X$; for $f : X \longrightarrow Y$, the notation $f(x)$ may be used to indicate the U-element $x \circ f \in Y$.

The *product* of a pair of objects X, Y in $\mathbf{T}$ is a further object $X \times Y$ endowed with two arrows $\pi_X : X \times Y \longrightarrow X$ and $\pi_Y : X \times Y \longrightarrow Y$, such that for every pair of U-elements $x : U \longrightarrow X$, $y : U \longrightarrow Y$, there exists a unique U-element $\langle x, y\rangle : U \longrightarrow X \times Y$ such that $\pi_X(\langle x, y\rangle) = x$ and $\pi_Y(\langle x, y\rangle) = y$. Thus this definition is a way of saying that '$X \times Y$ is formed by the set of ordered pairs of elements from X and Y', like a honest set-theoretic cartesian product. Similarly, a *terminal object* $\mathbf{1}$ in $\mathbf{T}$ is any object having just one U-element (for every U). By a *cartesian* category, we mean a category with products of pairs of objects and a terminal object. In a cartesian category, by iteration, it is possible to define the product of $X_1 \times \cdots \times X_n$ of n objects $X_1, \ldots, X_n$. For $n = 0$, $X_1 \times \cdots \times X_n$ is the terminal object $\mathbf{1}$ and, whenever $X_1 = \cdots = X_n := X$, the product $X_1 \times \cdots \times X_n$ is noted as X^n and called the n-th cartesian power of X.

Given arrows $Y_1 \xrightarrow{f_1} X \xleftarrow{f_2} Y_2$, the pullback P of f_1, f_2 is 'formed by the pairs' of U-elements $y_1 \in Y_1, y_2 \in Y_2$ such that $f_1(y_1) = f_2(y_2)$. This means that we have a commutative square

[15] The technique of generalized elements is common folklore in the category-theoretic community; we nevertheless wish to thank A. Carboni for pointing out the opportunity of introducing it at a very early stage.

$$\begin{tikzcd} P \arrow[r, "p_1"] \arrow[d, "p_2"'] & Y_1 \arrow[d, "f_1"] \\ Y_2 \arrow[r, "f_2"'] & X \end{tikzcd}$$

such that, for all U-elements $y_1 \in Y_1, y_2 \in Y_2$, if $f_1(y_1) = f_2(y_2)$, then there is a unique $y : U \longrightarrow P$ such that $p_1(y) = y_1$ and $p_2(y) = y_2$. A *left exact* (or just lex) *category* is a category with pullbacks, terminal object and products (actually, requiring products turns out to be redundant).

As happens with all category-theoretic notions, products, terminal objects, pullbacks, etc. are uniquely determined only *up to isomorphism* (where an isomorphism among objects X, Y is a pair of arrows $f : X \longrightarrow Y, g : Y \longrightarrow X$ such that $f \circ g = id_Y$ and $g \circ f = id_X$).

An arrow $m : A \hookrightarrow X$ is mono iff 'it is injective', i.e., iff $f(a_1) = f(a_2)$ implies $a_1 = a_2$, for every pair of U-elements of A. If x is a U-element of X and $m : A \hookrightarrow X$ is mono, we say that $x \in_X A$ iff[16] there is $a : U \longrightarrow A$ such that $m(a) = x$. We can compare monos $A_1 \hookrightarrow X$, $A_2 \hookrightarrow X$ by defining a suitable 'inclusion': formally, $A_1 \leq A_2$ holds iff for every U, for every U-element $x \in X$ ($x \in_X A_1 \Rightarrow x \in_X A_2$). Moreover $A_1 \sim A_2$ holds iff we have both $A_1 \leq A_2$ and $A_2 \leq A_1$; a *subobject* of X is then defined as an equivalence class (under $\sim$) of monos of codomain X. The relation $\leq$ gives the set of subobjects of X a partial order structure $(Sub(X), \leq)$: in fact, reflexivity, transitivity and antisymmetry of $\leq$ are immediately checked through generalized elements.

It is easily seen that in a pullback diagram

$$\begin{tikzcd} f^{-1}(A) \arrow[r] \arrow[d, tail] & A \arrow[d, tail, "m"] \\ Y \arrow[r, "f"'] & X \end{tikzcd}$$

the left vertical arrow is mono in case the right vertical arrow is mono. Notice that $f^{-1}(A)$, as a subobject of Y, is precisely inverse image, in the sense that we have $\forall y \in Y(y \in_Y f^{-1}(A) \Leftrightarrow f(y) \in_X A)$. Relations like

$$A \leq B \Rightarrow f^{-1}(A) \leq f^{-1}(B), \qquad id_X^{-1}(A) = A \qquad (f \circ g)^{-1}(A) = g^{-1}(f^{-1}(A))$$

for $A, B \in Sub(X)$ and $Z \xrightarrow{g} Y \xrightarrow{f} X$ can be easily established by using U-elements again. Such relations say that, for a given category $\mathbf{T}$, the correspondence associating with X the partial order $(Sub(X), \leq)$ and with $f : Y \longrightarrow X$ the order-preserving map $f^{-1} : (Sub(Y), \leq) \longrightarrow (Sub(X), \leq)$ is in fact a contravariant functor (see below) from $\mathbf{T}$ to the category of partial orders and order-preserving maps. This functor is called the *subobject functor* and plays a central role in the categorical analysis of logic.

If $\mathbf{T}$ is lex, it is easily seen that the subobject functor takes values into the category of semilattices and related morphisms (the meet of $A_1 \hookrightarrow X$, $A_2 \hookrightarrow X$ is their pullback). For rich $\mathbf{T}$, subobjects have further structure: for instance, in **Set**, subobjects are Boolean

[16] It is customary, by abuse, to name a mono $m : A \hookrightarrow X$ by A instead of by m (often m is not even mentioned).

algebras, moreover not only inverse but also direct image of a subobject (=subset) along an arrow (=function) can be defined. Understanding these richer structures is the goal of categorical logic and can be achieved mainly in two ways. The first (even from a historical point of view) approach is perhaps more empirical, but more flexible: this is the *hyperdoctrinal* method which tries to abstractly axiomatize the desired properties of the pair (category $\mathbf{T}$, subobject functor). The second approach, the *logical categories* approach, is exemplified in the book [85] and tries to understand what conditions on $\mathbf{T}$ itself lead precisely the subobject functor (not just an abstract 'subobject-like' functor) to have the desired structure. The philosophical effect of the logical categories program, if accomplished, is that of *explaining all the logical constructors* in terms of properties of the notion of composition of two arrows in a category. In the sequel, we take the hyperdoctrinal point of view, but we report also the achievements [84] of the logical categories analysis of modal logic.

We conclude this section by giving the definition of a *functor* $F : \mathbf{T_1} \longrightarrow \mathbf{T_2}$ between categories $\mathbf{T_1}$ and $\mathbf{T_2}$: this is a correspondence associating with every object X in $\mathbf{T_1}$ an object $F(X)$ in $\mathbf{T_2}$ and with every arrow $f : Y \longrightarrow X$ in $\mathbf{T_1}$ an arrow $F(f) : F(Y) \longrightarrow F(X)$ in $\mathbf{T_2}$. Identity and composition should be preserved, in the sense that we must have:

$$F(id_X) = id_{F(X)}, \qquad F(f \circ g) = F(f) \circ F(g),$$

for $Z \xrightarrow{g} Y \xrightarrow{f} X$ in $\mathbf{T_1}$. The functor F is said to preserve products, terminal objects, pullbacks iff it sends a product, terminal object, pullback diagram to a product, terminal object, pullback diagram, respectively. A *cartesian functor* is a functor among cartesian categories preserving products and terminal object, whereas a *left exact* (or just lex) *functor* is a cartesian functor among lex categories preserving also pullbacks. Notice that, since an arrow m is mono iff the pullback of m with itself is the identity, a lex functor preserves monos (and also the partial order among subobjects). Finally, a contravariant functor from $\mathbf{T_1}$ to $\mathbf{T_2}$ is a functor $F : \mathbf{T_1} \longrightarrow \mathbf{T_2}^{op}$, where $\mathbf{T_2}^{op}$ (the opposite category of $\mathbf{T_2}$) is the category obtained from $\mathbf{T_2}$ by 'reversing arrow directions'.

9.3 Hyperdoctrines

Hyperdoctrines were introduced by F.W. Lawvere in [78]; they were adapted to the counterpart semantics of Section 11 in [55], [56] and to standard first-order non classical logics in [114]. Recall that *interior algebras* are the algebraic models of propositional **S4**: an interior algebra is a Boolean algebra $(B, \wedge, \vee, \top, \bot, \neg)$ endowed with an operator $\Box$ satisfying the equations $\Box(a \wedge b) = \Box a \wedge \Box b$, $\Box\top = \top$, $\Box\Box a = \Box a$, $\Box a \leq a$ (recall that inequations like $c \leq d$ are defined as $c \wedge d = c$ in lattices). Interior algebras and the appropriate morphisms form the category **Int**.

DEFINITION 40. A *modal hyperdoctrine*, based on the cartesian category $\mathbf{T}$, is a contravariant functor $\mathcal{A} : \mathbf{T} \longrightarrow \mathbf{Int}^{op}$ satisfying the following two requirements.

(i) For any arrow $f : Y \longrightarrow X$ in $\mathbf{T}$, the 'inverse image' morphism $\mathcal{A}(f) : \mathcal{A}(X) \longrightarrow \mathcal{A}(Y)$ has a left adjoint (the 'direct image along f') $\exists_f : \mathcal{A}(Y) \longrightarrow \mathcal{A}(X)$ and a right adjoint (the 'dual image along f') $\forall_f : \mathcal{A}(Y) \longrightarrow \mathcal{A}(X)$.[17] This means that the

[17] As we are in a Boolean context, these adjoints are interdefinable, e.g. we have $\exists_f(A) = \neg\forall_f(\neg A)$.

following conditions must be satisfied for all $A \in \mathcal{A}(Y)$ and $B \in \mathcal{A}(X)$ (we write f^{-1} instead of $\mathcal{A}(f)$):

$$\begin{array}{ll} B \leq f^{-1}(A) \Rightarrow \exists_f(B) \leq A, & B \leq f^{-1}(\exists_f(B)) \\ f^{-1}(A) \leq B \Rightarrow A \leq \forall_f(B), & f^{-1}(\forall_f(B)) \leq B. \end{array}$$

(ii) The Beck-Chevalley condition holds; this means that for every term/projection pullback, i.e., for every pullback of the kind[18]

$$\begin{tikzcd} Y \times Z \arrow[r, "f \times id_Z"] \arrow[d, "\pi_Y"'] & X \times Z \arrow[d, "\pi_X"] \\ Y \arrow[r, "f"'] & X \end{tikzcd}$$

the following two identities are satisfied for all $A \in \mathcal{A}(X \times Z)$:

$$f^{-1}(\exists_{\pi_X}(A)) = \exists_{\pi_Y}((f \times id_Z)^{-1}(A)), \qquad f^{-1}(\forall_{\pi_X}(A)) = \forall_{\pi_Y}((f \times id_Z)^{-1}(A)).$$

The meaning of Definition 40 is better explained through the corresponding Lindenbaum construction: given a language $\mathcal{L}$, we take as $\mathbf{T}_{\mathcal{L}}$ the category whose objects are natural numbers $n \geq 0$ and whose arrows $n \longrightarrow m$ are m-tuples of n-terms (composition is substitution). For a given modal first-order system $\mathbf{S}$, let $\mathcal{A}_{\mathbf{S}}(n)$ be the set of equivalence classes of n-formulas (equivalence is meant with respect to provability of bi-implication in $\mathbf{S}$). For an m-tuple of n-terms $\underline{t} = (t_1, \ldots, t_m) : n \longrightarrow m$, the inverse image interior algebras morphism $\mathcal{A}(\underline{t})$ maps the equivalence class of $\psi : m$ to the equivalence class of $\psi[\underline{t}] : n$. Moreover $\exists_{\underline{t}}$ is 'syntactic direct image', that is the operation of taking the equivalence class of a formula $\phi : n$ to the equivalence class of the formula (let z_i be x_{m+i})

$$\exists z_1 \cdots \exists z_n (\phi[z_1, \ldots, z_n] \wedge \bigwedge_{i=1}^{m} t_i[z_1, \ldots, z_n] = x_i) : m.$$

Thus, when $n = m+1$ and $\underline{t} = x_1, \ldots, x_m$, $\exists_{\underline{t}}$ applied to $\phi : m+1$ is just the equivalence class of $\exists x_{m+1}\phi : m$. Beck-Chevalley conditions hold because of Definition 39(c), hence it is easily seen that $\mathcal{A}_{\mathbf{S}}$ is a modal hyperdoctrine over $\mathbf{T}_{\mathcal{L}}$ (called the 'Lindenbaum hyperdoctrine' of $\mathbf{S}$).

In fact, there is a sense in which every (small) modal hyperdoctrine is equivalent to a Lindenbaum hyperdoctrine (of a multi-sorted theory, rather than of a system). Thus (small) modal hyperdoctrines may be seen as syntactic, rather than semantic frameworks. This is why the following definition of a hyperdoctrinal model becomes really significant only after the next subsections introduction of suitable special 'semantic' modal hyperdoctrines.

DEFINITION 41. Let a language $\mathcal{L}$ be given; a hyperdoctrinal model $\mathfrak{M} = (\mathbf{T}, \mathcal{A}, D, \mathcal{I})$ consists of the following data:

- a modal hyperdoctrine $\mathcal{A}$ over the cartesian category $\mathbf{T}$;

[18]The arrow $f \times id_Z$ is defined as $\langle f \circ \pi_Y, \pi_Z \rangle$.

- an object D of $\mathbf{T}$ (the domain of $\mathfrak{M}$);
- an interpretation $\mathcal{I}$ mapping every function symbol f to an arrow $\mathcal{I}(f) : D^n \longrightarrow D$ of $\mathbf{T}$ and every predicate symbol P to an element $\mathcal{I}(P) \in \mathcal{A}(D^n)$ (here n is the arity of f, P).

In Definition 41, it is assumed also that $\mathcal{I}(\bot) := \bot$ and that $\mathcal{I}(=) := \exists_{\Delta_D}(\top)$, where Δ_D is the diagonal map $\langle id_D, id_D\rangle : D \longrightarrow D^2$.

DEFINITION 42. Given a hyperdoctrinal model $\mathfrak{M} = (\mathbf{T}, \mathcal{A}, D, \mathcal{I})$ and an n-term $t : n$, the term $[\![t]\!]^{\mathfrak{M}} : D^n \longrightarrow D$ is so defined by induction:

- if $t = x_i$ $(i = 1, \dots, n)$, then $[\![t]\!]^{\mathfrak{M}}$ is the i-th projection $\pi_i : D^n \longrightarrow D$;
- if $t = f(t_1, \dots, t_k)$, then $[\![t]\!]^{\mathfrak{M}}$ is given by the composition

$$D^n \xrightarrow{\langle [\![t_1]\!]^{\mathfrak{M}}, \dots, [\![t_k]\!]^{\mathfrak{M}}\rangle} D^k \xrightarrow{\quad \mathcal{I}(f) \quad} D.$$

DEFINITION 43. Given a hyperdoctrinal model $\mathfrak{M} = (\mathbf{T}, \mathcal{A}, D, \mathcal{I})$ and an n-formula $\phi : n$, we define $[\![\phi]\!]^{\mathfrak{M}} \in \mathcal{A}(D^n)$ as follows, by induction:

(i) if $\phi : n$ is $P(t_1, \dots, t_k)$, then $[\![\phi]\!]^{\mathfrak{M}} := \langle [\![t_1]\!]^{\mathfrak{M}}, \dots, [\![t_k]\!]^{\mathfrak{M}}\rangle^{-1}(\mathcal{I}(P))$;

(ii) if $\phi : n$ is $\phi_1 \to \phi_2 : n$, then $[\![\phi]\!]^{\mathfrak{M}} := [\![\phi_1]\!]^{\mathfrak{M}} \to [\![\phi_2]\!]^{\mathfrak{M}}$ (this is the Boolean relative complement operation in $\mathcal{A}(D^n)$);

(iii) if $\phi : n$ is $\forall y \psi : n$, then $[\![\phi]\!]^{\mathfrak{M}} := \forall_{\langle \pi_1, \dots, \pi_n\rangle}([\![\psi]\!]^{\mathfrak{M}})$ (this is the dual image along the n-tuple of the first n projections of domain D^{n+1} and of codomain D);

(iv) if $\phi : n$ is $\Box\psi : n$, then $[\![\phi]\!]^{\mathfrak{M}} := \Box[\![\psi]\!]^{\mathfrak{M}}$ (this is the Box operator in the interior algebra $\mathcal{A}(D^n)$).

An n-formula $\phi : n$ is *valid in* $\mathfrak{M}$ iff $[\![\phi]\!]^{\mathfrak{M}} = \top$.

We can now prove that the system **QS4** of Table 1 is sound with respect to hyperdoctrinal models. The proof is indeed simple (it would be much more tedious for untyped systems); we first need an instantiation lemma, which is immediate by Definition 39, by the fact that inverse image in a modal hyperdoctrine is an interior algebra morphism and by Beck-Chevalley condition:

LEMMA 44. *In every modal hyperdoctrinal* $\mathfrak{M}$, *we have*

$$[\![\phi[t_1, \dots, t_k]]\!]^{\mathfrak{M}} = \langle [\![t_1]\!]^{\mathfrak{M}}, \dots, [\![t_k]\!]^{\mathfrak{M}}\rangle^{-1}([\![\phi]\!]^{\mathfrak{M}}),$$

for every k-formula $\phi : k$ and for every k-tuple of n-terms $t_1 : n, \dots, t_k : n$.

THEOREM 45. *(Validity) If the n-formula $\phi : n$ is provable in* **QS4**, *then it is valid in every hyperdoctrinal model* $\mathfrak{M}$.

Proof. The **S4**-propositional tautologies and the rules (MP) and (Nec) are valid by interior algebras axioms; the instantiation rule $(Inst)$ is valid by Lemma 44. The rule $(\forall\text{-}In)$ and the axiom schemata $(\forall\text{-}Ex)$ are just logical translations of the adjointness

conditions. Take for instance the case of ($\forall$-*Ex*): by Lemma 44, we have $[\![(\forall y\, \phi)[\underline{x}]]\!]^{\mathfrak{M}} = ([\![\underline{x}]\!]^{\mathfrak{M}})^{-1}([\![\forall y \phi]\!]^{\mathfrak{M}})$, which is equal to $\langle \pi_1, \ldots, \pi_n\rangle^{-1}(\forall_{\langle \pi_1, \ldots, \pi_n\rangle}([\![\phi]\!]^{\mathfrak{M}}))$; this is less or equal to $[\![\phi]\!]^{\mathfrak{M}}$ by adjointness.

We finally justify the validity of the equality axiom schemata. Now $[\![x_1 = x_2]\!]^{\mathfrak{M}}$ is $\exists_{\Delta_D}(\top)$, hence $[\![x_1 = x_1]\!]^{\mathfrak{M}}$ is equal to $\Delta_D^{-1}(\exists_{\Delta_D}(\top))$, i.e., to $\top$ by adjointness. Similarly, the axiom schemata (*Rep*) is equivalent (using the remaining rules and axioms of Table 1) to

$$x_1 = x_2 \to \forall x_3 \cdots \forall x_{n+2}(\phi[x_1, x_3, \ldots, x_{n+2}] \to \phi[x_2, x_3, \ldots, x_{n+2}]) : 2.$$

Now, by adjointness, we can move $\exists_{\Delta_D}$ in the antecedent of the implication to $\Delta_D^{-1} = \langle [\![x_1]\!]^{\mathfrak{M}}, [\![x_1]\!]^{\mathfrak{M}}\rangle^{-1}$ in the consequent, apply instantiation Lemma 44, Definition 39(c) and get a tautology. ❑

Let $\mathcal{K}$ be a class of modal hyperdoctrines (e.g. the class of modal hyperdoctrines arising from metaframes, presheaves, etc., see next subsections). We say that a modal system $\mathbf{S}$ is *complete with respect to $\mathcal{K}$-semantics* (or simply with respect to $\mathcal{K}$) iff every formula which is not provable in $\mathbf{S}$ fails in a hyperdoctrinal model $\mathfrak{M} = (\mathbf{T}, \mathcal{A}, D, \mathcal{I})$, such that: (i) $(\mathbf{T}, \mathcal{A}) \in \mathcal{K}$; (ii) all formulas provable from $\mathbf{S}$ are valid not only in $\mathfrak{M}$, but also in every model of the kind $\mathfrak{M} = (\mathbf{T}, \mathcal{A}, D, \mathcal{I}')$ (varying $\mathcal{I}'$).[19]

Thus, for instance, every modal system $\mathbf{S}$ is complete with respect to the class $\mathcal{K}$ formed by all the modal hyperdoctrines [114]: to prove it, it is sufficient to work in the Lindenbaum hyperdoctrine of $\mathbf{S}$.

However, given an arbitrary hyperdoctrine $(\mathbf{T}, \mathcal{A})$ and an object D from $\mathbf{T}$, it is not always the case that the set of formulas valid in all the models of the kind $\mathfrak{M} = (\mathbf{T}, \mathcal{A}, D, \mathcal{I})$ is a modal system. This set of formulas is closed only under modus ponens, generalization, necessitation and *exact* uniform substitutions.[20] This phenomenon is common to most of the extended semantics proposed in the literature (such as Kripke bundles, presheaves or metaframes): thus, in order to define *the modal system of* $(\mathbf{T}, \mathcal{A}, D)$, one should take the set of formulas whose substitution instances are all valid in the models of the kind $(\mathbf{T}, \mathcal{A}, D, \mathcal{I})$.

9.4 Metaframes

Metaframes [121] can be easily introduced (and understood) through the hyperdoctrinal approach [114]. Notice first that preordered sets and *open* maps form a category $\mathbf{F}$. Here a map $f : (W, \leq) \longrightarrow (W', \leq')$ is said to be open iff it is order-preserving and moreover satisfies the condition $f(w) \leq' w' \;\Rightarrow\; \exists v\,(w \leq v \;\&\; f(v) = w')$, for all $w \in W, w' \in W'$. For a given preordered set $\mathfrak{F} = (W, \leq)$, the powerset Boolean algebra $\mathcal{P}(W)$ can be turned into an interior algebra by defining, for $S \subseteq W$, $\Box S = \{w \in W \mid \forall v\,(w \leq v \;\Rightarrow\; v \in S)\}$. Similarly, taking inverse image along an open map $f : (W, \leq) \longrightarrow (W', \leq')$ is an interior algebras morphism $\mathcal{P}(f) := f^{-1} : \mathcal{P}(W') \longrightarrow \mathcal{P}(W)$. In this way, we actually defined a contravariant functor $\mathcal{P} : \mathbf{F} \longrightarrow \mathbf{Int}^{op}$.

Given a cartesian category $\mathbf{T}$, a $\mathbf{T}$-*metaframe* is a functor $M : \mathbf{T} \longrightarrow \mathbf{F}$ such that the composite functor $\mathcal{A}_M : \mathbf{T} \xrightarrow{M} \mathbf{F} \xrightarrow{\mathcal{P}} \mathbf{Int}^{op}$ is a modal hyperdoctrine over $\mathbf{T}$.

[19] A stronger (yet unexplored) variant of this definition would allow also D to vary.

[20] An exact uniform substitution [121] of $\phi : n$ is a formula of the kind $\phi[\psi/P] : n$, where $\psi : k$ is a k-formula replacing the predicate symbol P of arity k.

This definition can be equivalently stated directly in terms of the functor $M : \mathbf{T} \longrightarrow \mathbf{F}$ as follows. Notice that set-theoretic inverse image along f always has both adjoints (namely, direct image and dual image), so in order for M to be a $\mathbf{T}$-metaframe, only the Beck-Chevalley condition has to be considered. However, in this setting, f^{-1}, $\exists_f$ and $\forall_f$ can be conveniently treated as modal (two-sorted) operators and, as Beck-Chevalley condition is in Sahlqvist form, standard correspondence machinery applies. Thus it is not difficult to see that M is a $\mathbf{T}$-metaframe iff for the term-projections pullbacks of Definition 40(ii), the following 'lifting' condition is satisfied, for all $a \in M(X \times Z), b \in M(Y)$:

$$M(\pi_X)(a) = M(f)(b) \;\Rightarrow\; \exists c \in M(Y \times Z)\; (M(f \times id)(c) = a \;\&\; M(\pi_Y)(c) = b).$$

If M is a $\mathbf{T}$-metaframe, a *metaframe model* on M can be defined as a hyperdoctrinal model over $(\mathbf{T}, \mathcal{A}_M)$,[21] so that all the semantic definitions of the previous subsection apply to metaframes as special cases. Metaframes define a powerful semantics, as is shown by the following result:

THEOREM 46. [121] *Let* **L** *be a canonical propositional modal logic. Then:*

(i) **QL** *is complete in metaframes semantics;*

(ii) **BF**.**QL** *is complete in metaframes semantics.*

The proof essentially uses the composition of the Lindenbaum functor $\mathcal{A}_{\mathbf{QL}}$ with the 'canonical frame functor': notice, in fact, that propositional axioms transfer to canonical frames by assumption and that both Beck-Chevalley condition and the Barcan formula are 'in Sahlqvist form', so they transfer too.

The distinguishing feature of modal metaframes semantics is that 'products are not preserved', i.e., for a $\mathbf{T}$-metaframe M, we have that $M(X \times Y)$ is not even a subset of $M(X) \times M(Y)$: this has the consequence that a, say binary, predicate symbol is not interpreted as a set of pairs of individuals, but just as a set of 'abstract pairs'.

One may wonder whether Theorem 46 holds for other ('product-preserving') semantics: by applying saturated model-theory techniques, [53] *proves Theorem 46(i) for presheaf semantics*[22] in the case of intermediate logics. The related extension to **S4**-modal logics holds, provided **L** is assumed to be not only canonical, but also closed under a 'cluster-expansion' semantic condition (see [53] for details)

10 MATHEMATICAL MODELS FOR MODALITIES

The aim of this section is to show that modalities naturally arise in well-established mathematical frameworks. The related analysis will give us new insight for the axiomatization of D. Lewis' counterpart semantics.

[21] Taking into consideration the special case in which the language does not contain constant or function symbols, $\mathbf{T}$ is (equivalent to) the opposite of the category of finite sets and the domain of the model is a singleton set, this substantially agrees with the definition of [121] (but our interpretation of the equality predicate in a metaframe model is different).

[22] This is the semantics explained in the Example 4 of Subsection 10.1 below.

10.1 Geometric Morphisms and Modal Logic

An (elementary) topos $\mathbf{E}$ is a lex cartesian closed category with subobject classifier. In the sequel, we shall directly mention the relevant consequences that can be drawn from this definition that matter for our purpose of interpreting first-order systems in a topos. For the moment, let us just mention a couple of examples of toposes that are used in intensional semantics.

Example 1. Kripke-like universes: fix a preordered Kripke frame $\mathfrak{F} = (W, \leq)$. A presheaf on $\mathfrak{F}$ is a contravariant functor into the category $\mathbf{Set}$ of sets. That is, D associates with $w \in W$ a set D_w (the set of 'individuals' living in it) and, whenever the relation $v \leq w$ holds, D_{vw} is a function $D_w \longrightarrow D_v$; such data should satisfy the requirements $D_{ww} = id$ and $D_{vw} \circ D_{wz} = D_{vz}$. These presheaves on preordered sets (investigated also in some textbooks on intuitionistic logic like [27]) are sometimes called *Kripke frames with equality*: they differ from the standard Kripke $\mathfrak{F}$-domains of Section 6 because the inclusion $D_w \subseteq D_v$ is replaced by an arbitrary function (we also reversed the direction of such a function in accordance with current topos-theoretic literature).[23]

A natural transformation $f : D \longrightarrow D'$ among our presheaves is a collection of functions $\{f_w : D_w \longrightarrow D'_w \mid w \in W\}$ such that for $v \leq w$ the square

$$\begin{tikzcd} D_w \arrow[r, "f_w"] \arrow[d, "D_{vw}"'] & D'_w \arrow[d, "D'_{vw}"] \\ D_v \arrow[r, "f_v"'] & D'_v \end{tikzcd}$$

commutes.[24] Presheaves and natural transformations are the 'Kripkean' topos $\mathbf{Set}^{\mathfrak{F}^{\mathbf{op}}}$.

The generalization to the case in which $\mathfrak{F}$ is a category (and not simply a preordered set) is straightforward: if $\mathfrak{F}$ is a category, a presheaf on it is a contravariant functor from $\mathfrak{F}$ to $\mathbf{Set}$, a morphism among such presheaves is a natural transformation (=family of maps indexed by the objects of $\mathfrak{F}$ making the obvious squares to commute), these presheaves and natural transformations are a category which is a topos, the latter is called $\mathbf{Set}^{\mathfrak{F}^{\mathbf{op}}}$ again. Usually, in the literature, when people deal with presheaf semantics, they actually refer to *presheaves over an arbitrary category.* Notice also that functional frames of Section 8 can be easily turned into presheaves on a category.

Example 2. Etale spaces: fix a topological space S. A local homeomorphism $e : E \longrightarrow S$ (or etale space, or sheaf over S) is a continuous map among topological spaces such that for every $a \in E$ there are open neighborhoods N of a and M of $e(a)$ such that e restricts to a homeomorphism $e_{|N} : N \longrightarrow M$. A map among etale spaces $e : E \longrightarrow S$ and $e' : E' \longrightarrow S$ is a continuous map $f : E \longrightarrow E'$ such that $e' \circ f = e$. Etale spaces and related maps are the topos $Sh(S)$.

Recall the subobject functor Sub from Subsection 9.2: this functor associates with an object X in a topos $\mathbf{E}$ the equivalence classes of monos $A \hookrightarrow X$; moreover, for every arrow $f : Y \longrightarrow X$ in $\mathbf{E}$, $Sub(f) = f^{-1} : Sub(X) \longrightarrow Sub(Y)$ takes pullback. As

[23] As pointed out in [111], Kripke frames with equality are semantically stronger than standard Kripke $\mathfrak{F}$-domains, even for systems without equality.

[24] In case $D = \mathbf{1}$ is the one-point terminal presheaf, the commutativity of the square just expresses a kind of 'rigid designator' condition for the global constant $f : \mathbf{1} \longrightarrow D'$.

the categorical structure of a topos is quite rich, we have the following lemma (see any textbook like [83], [70] for the proof):

LEMMA 47. *Every topos* $\mathbf{E}$ *is a Heyting category, meaning that the subobject functor*

$$Sub : \mathbf{E} \longrightarrow \mathbf{Heyt}^{op}$$

endows $\mathbf{E}$ *with an intuitionistic hyperdoctrinal structure.*

The notion of an intuitionistic (resp. classical) hyperdoctrine, is obtained from Definition 40 by replacing in it the category **Int** by the category **Heyt** of Heyting algebras (resp. by the category **Bool** of Boolean algebras). Whenever the intuitionistic hyperdoctrinal structure mentioned in Lemma 47 is in fact a classical hyperdoctrinal structure, the topos $\mathbf{E}$ is said to be *Boolean*.

In order to extend the above analysis from intuitionistic to modal logic, we must first understand in a general topos-theoretic context an evident anomaly that characterizes Kripke semantics for first-order modal logic (when compared with the corresponding semantics for intuitionistic logic). In first-order Kripke semantics for modal logic, sorts are interpreted as presheaves and terms are interpreted as natural transformations (at least in the rigid designators case). What is peculiar is the interpretation of predicates: in fact a, say unary, predicate P is interpreted as a collection of subsets $\{P_w \subseteq D_w \mid w \in W\}$ which is *not* a subpresheaf (i.e., it is not a subobject of D in the topos $\mathbf{Set}^{\mathfrak{F}^{\mathbf{op}}}$). Thus, the categorical structure of a topos seems to be unable to give a full account of modal logic. There is however a quite simple and beautiful solution to this problem for our **S4** systems: we only need to consider *two* toposes and a geometric morphism connecting them.[25]

The most obvious notion of a morphism $F : \mathbf{E}_1 \longrightarrow \mathbf{E_2}$ among toposes is the so-called 'logical' notion: F is taken to be a functor that preserves the topos structure (i.e., finite limits, exponentials and subobject classifier). However, geometry suggests another notion, which is the good one for modal logic too. In fact, a continuous map $f : S \longrightarrow T$ among topological spaces, induces two functors $F_* : Sh(S) \longrightarrow Sh(T)$ and $F^* : Sh(T) \longrightarrow Sh(S)$ such that (i) F^* is left adjoint to F_*;[26] (ii) F^* is left exact. Thus, let us define a *geometric morphism* $\mathbf{E}_1 \longrightarrow \mathbf{E_2}$ among toposes $\mathbf{E}_1, \mathbf{E}_2$ as a pair of functors $F_* : \mathbf{E}_1 \longrightarrow \mathbf{E}_2$ and $F^* : \mathbf{E}_2 \longrightarrow \mathbf{E}_1$ such that F^* is left exact left adjoint to F_*.

Example 3. This definition gives what we are looking for in the Kripkean case. Let $f : \mathfrak{G} \longrightarrow \mathfrak{F}$ be an order-preserving maps among preordered sets (again the generalization to the case in which $\mathfrak{F}, \mathfrak{G}$ are categories and f is a functor, is straightforward). Such an f induces a functor $F^* : \mathbf{Set}^{\mathfrak{F}^{\mathbf{op}}} \longrightarrow \mathbf{Set}^{\mathfrak{G}^{\mathbf{op}}}$, just by 'taking composition with f'. It is well-known (from the theory of Kan extensions [82]) that F^* has both a right adjoint F_* and a left adjoint $F_!$, hence the pair (F_*, F^*) defines a geometric morphism $\mathbf{Set}^{\mathfrak{G}^{\mathbf{op}}} \longrightarrow \mathbf{Set}^{\mathfrak{F}^{\mathbf{op}}}$. For Kripke semantics, the relevant special case is the case in which $\mathfrak{G}$ is the discrete poset

[25] This idea seems to be due to F. W. Lawvere and was developed in a series of papers by G. E. Reyes and other people [103], [77], [104], [84] (we substantially follow [84] in our exposition). Topos-theoretic semantics *always validates* **S4** *axioms* (see Lemma 48(i)), hence it is unsuitable for weaker systems; in contrast to all the other semantics that are introduced in this chapter, there is no evident way of generalizing the mechanism of topos-theoretic semantics so that it can be adapted to weaker normal systems.

[26] Adjointness means that there is a natural bijection (called transposition) among the hom-sets $Hom(F^*(X), Y) \simeq Hom(X, F_*(Y))$, for every object X in the domain category of F^* and for every object Y in the domain category of F_*.

$(W,=)$ formed by the set of possible words of $\mathfrak{F} = (W, \leq)$. In this case, if $f : (W,=) \longrightarrow \mathfrak{F}$ is the identity map, $F^* : \mathbf{Set}^{\mathfrak{F}^{\mathbf{op}}} \longrightarrow \mathbf{Set}^{(W,=)}$ is the functor associating with the presheaf D, the same D seen as the collection of sets $\{D_w \mid w \in W\}$ indexed by the possible worlds (otherwise said, the maps D_{vw} are forgotten). Subobjects of $F^*(D)$ in the topos $\mathbf{Set}^{(W,=)}$ are consequently what is needed in order to interpret predicates in ordinary modal Kripke semantics.

How to recover modal operators from a geometric morphism $(F_*, F^*) : \mathbf{E}_1 \longrightarrow \mathbf{E}_2$ in the general case? This is done as follows: take an object D from $\mathbf{E}_2$ and a subobject $A \hookrightarrow F^*(D)$. We apply F_* (which preserves monos, being a right adjoint) and get first $F_*(A) \hookrightarrow F_*(F^*(D))$. Now consider the unity $\eta_D : D \longrightarrow F_*(F^*(D))$ of the adjunction (this is the transpose of the identity map on $F^*(D)$) and take the pullback

$$\begin{tikzcd}
\flat(A) \arrow[r, tail] \arrow[d] & D \arrow[d, "\eta_D"] \\
F_*(A) \arrow[r, tail] & F_*(F^*(D))
\end{tikzcd}$$

Applying F^* again (this functor preserves monos, as it is left exact) we finally get a subobject $F^*(\flat(A)) \hookrightarrow F^*(D)$, which we call $\Box A$. Thus we defined operators

$$\flat : Sub(F^*(D)) \longrightarrow Sub(D) \quad \text{and} \quad \Box : Sub(F^*(D)) \longrightarrow Sub(F^*(D)),$$

whose relevant properties are summarized in the following lemma (we omit proofs, that can be easily obtained by unravelling the definitions):

LEMMA 48.

(i) *For $A, B \in Sub(F^*(D))$, we have*

$$\Box A \leq A, \qquad \Box A = \Box\Box A,$$
$$\Box id_{F^*(D)} = id_{F^*(D)}, \qquad \Box(A \wedge B) = \Box A \wedge \Box B,$$

that is, $Sub(F^*(D))$ is an (intuitionistic) interior algebra.

(ii) *For every $h : D' \longrightarrow D$ in $\mathbf{E}_2$ and for every $A \in Sub(F^*(D))$, we have*

$$\Box(F^*(h))^{-1}(A) = (F^*(h))^{-1}(\Box A),$$

that is, $(F^*(h))^{-1}$ is an interior algebra morphism.

(iii) *$\flat$ is the right adjoint to the operator $Sub(D) \longrightarrow Sub(F^*(D))$ taking $A \hookrightarrow D$ to $F^*(A) \hookrightarrow F^*(D)$. This means that we have*

$$F^*(B) \leq A \Rightarrow B \leq \flat(A), \qquad F^*(\flat(A)) \leq A,$$

for all $B \in Sub(D), A \in Sub(F^*(D))$.

Conditions (i)-(ii) are the needed ingredients for the proof of next theorem, whereas condition (iii) makes easier the calculation of the $\Box$-operator in the relevant examples.

THEOREM 49. *Let* $(F^*, F_*) : \mathbf{E}_1 \longrightarrow \mathbf{E}_2$ *be a geometric morphism. Assume also that* $\mathbf{E}_1$ *is a Boolean topos.*[27] *Then the functor* $Sub \circ F^* : \mathbf{E}_2 \longrightarrow \mathbf{Int}^{op}$ *gives a modal hyperdoctrine over* $\mathbf{E}_2$*.*

Proof. Immediate by Lemmas 47 and 48(i)-(ii). ❑

Thus we can define a *topos-theoretic model* of $\mathcal{L}$ as a hyperdoctrinal model of $\mathcal{L}$ into a hyperdoctrine of the kind mentioned in Theorem 49.

Example 3 (continued). In the Kripke framework of Example 3, we can recover, from the definition of a topos-theoretic model, the standard Kripke forcing conditions for first-order modal logic in the following way. Given a presheaf D and a subobject $A \subseteq F^*(D)$ (that is, a collection of subsets $\{A_w \subseteq D_w \mid w \in W\}$), we do not need to know the complicated construction of F_* in order to compute $\flat(A) \subseteq D$: Lemma 48(iii) is sufficient for that. In fact, the adjointness conditions uniquely determine $\flat(A)$ (and hence also the collection of subsets $F^*(\flat A) = \Box A$) as the subpresheaf $\flat(A)_w = \{a \in D_w \mid \forall v\ (v \leq w \Rightarrow D_{vw}(a) \in A_v)\}$. Now, given a topos-theoretic model $\mathfrak{M}$ in the present Kripkean framework, for an n-formula $\phi : n$ and for an n-tuple $\underline{a} \in D^n_w$, write $\underline{a} \models^{\mathfrak{M}}_w \phi$ for $\underline{a} \in ([\![\phi]\!]^{\mathfrak{M}})_w$. Definition 43(iv) now reads[28]

$$\underline{a} \models^{\mathfrak{M}}_w \Box\psi \quad \text{iff} \quad \forall v\ (v \leq w \Rightarrow D_{vw}(\underline{a}) \models^{\mathfrak{M}}_v \psi)$$

(where $D_{vw}(\underline{a})$ means the componentwise application of the function D_{vw} to the tuple $\underline{a}$). Similarly, Definition 43(i)-(ii)-(iii) gives the standard forcing conditions for atomic formulas, for $\rightarrow$ and for $\forall$, respectively.

Example 4. If we allow $\mathfrak{F}$ to be a category in Example 3, we get full *presheaf semantics*, i.e., the semantics used in [52], [50] in order to prove some of the incompleteness results mentioned in Section 8. Forcing conditions for truth of modal formulas in full presheaf semantics (where presheaves are taken on a category) are derived in complete analogy to the Kripkean case of Example 3.[29] For powerful completeness results with respect to this semantics, see the final remark of Subsection 9.4.

Example 5. In a completely analogous way, forcing conditions can be derived in the etale case, thus establishing a suitable *topological semantics* for first-order modal logic. In fact, let S be a topological space and let $|S|$ be the discretization of S (points are the same as for S, but now every subset is open). The identity map $f : |S| \longrightarrow S$ is continuous and hence it induces a geometric morphism $(F_*, F^*) : Sh(|S|) \longrightarrow Sh(S)$. The 'inverse image part' F^* of this morphism associates with the etale space $e : D \longrightarrow S$ the discrete space $|e| : |D| \longrightarrow |S|$. Using Lemma 48(iii), the operator $\flat$ is easily seen to take a subset to its interior; notice also that (according to the characterization of products in the topos $Sh(S)$) D^n is now the fibered product of D with itself n-times.[30] Hence, given $p \in S$,

[27] This assumption is needed because we are studying (for simplicity) **S4**-systems on a classical basis; for modal systems on an intuitionistic basis this restriction is obviously dropped.

[28] Notice that here the n-tuple $\underline{a}$ plays the role of a *finitary* assignment to the variables $x_1, \ldots, x_n$.

[29] If functional frames are seen as presheaves, we get precisely the truth forcing conditions of Section 8 for functional models.

[30] This means that D^n is the subspace of the n-th cartesian power of D formed by the tuples $(a_1, \ldots, a_n)$ living on the same fiber (i.e., such that $e(a_1) = \cdots = e(a_n)$); the topology on D^n is the relativization of the product topology.

$a_1, \ldots, a_n \in e^{-1}(p)$, if we write $(a_1, \ldots, a_n) \models_p^{\mathfrak{M}} \phi$ for $(a_1, \ldots, a_n) \in [\![\phi]\!]^{\mathfrak{M}}$, we get from Definition 43(iv) the following topological forcing condition:

$$(T) \quad (a_1, \ldots, a_n) \models_p^{\mathfrak{M}} \Box\psi \quad \text{iff} \quad \begin{array}{l} \exists \text{ neighborhoods } J, I_1, \ldots, I_n \text{ of } p, \\ a_1, \ldots, a_n, \text{ respectively, s.t. } \forall q \in J, \\ \forall b_1, \ldots, b_n \in (I_1 \times \cdots \times I_n) \cap e^{-1}(q) \\ \text{we have that } (b_1, \ldots, b_n) \models_q^{\mathfrak{M}} \psi. \end{array}$$

When the topology in S is *preordered* (i.e., when there is a preorder relation on S such that open subsets are just the upward closed subsets), it is not difficult to see that we get again the Kripke frames with equality of Example 3 as a special case.

Validity of **QS4** with respect to topos-theoretic semantics follows from the general validity Theorem 45. Completeness of **QS4** can be proved as well: this however already follows from completeness with respect to Kripke frames with equality (a direct proof, based on the elegant Joyal construction for presheaf completeness of intuitionistic logic, is given in [84]).

10.2 The Need for the Continuity Axiom

We now briefly sketch some further semantic investigations into mathematically motivated models, raising new problems, whose solution requires a better understanding of the types mechanism (mainly in connection to substitution).

A geometric morphism $(F_*, F^*) : \mathbf{E}_1 \longrightarrow \mathbf{E_2}$ is called *essential* in case F^* has a further left adjoint $F_! : \mathbf{E}_1 \longrightarrow \mathbf{E_2}$ (as we saw in Example 3 from Subsection 10.1, this is always the case in Kripke-like frameworks). For essential geometric morphisms, we can define diamond operators $\Diamond_p : Sub(F^*(D)) \longrightarrow Sub(F^*(D))$ by a mechanism similar to the mechanism we used for the $\Box$-operators of Lemma 48. Given a subobject $A \hookrightarrow F^*(D)$, we apply $F_!$, compose with the counity of the adjointness

$$F_!(A) \longrightarrow F_!(F^*(D)) \longrightarrow D$$

and then take epi/mono factorization (which is available in any topos); in this way we get a subobject of D, which we turn into a subobject $\Diamond_p(S)$ of $F^*(D)$ by applying F^* once again. This operator $\Diamond_p : Sub(F^*(D)) \longrightarrow Sub(F^*(D))$ is not interdefinable (via negation) with the $\Box$-operator introduced in Subsection 10.1, rather it is left adjoint to it, meaning that the pair of operators $(\Diamond_p, \Box)$ jointly satisfy the axioms for tense logic (that is, $\Diamond_p$ is a 'past possibility' operator, whereas $\Box$ is a 'future necessity' operator). There is one point that goes wrong here, however: we fail to get a 'tense hyperdoctrine', because the statement of Lemma 48(ii) for $\Diamond_p$ is not true, we only have:

$$(C\Diamond_p) \qquad \Diamond_p(F^*(h))^{-1}(A) \leq (F^*(h))^{-1}(\Diamond_p A).$$

As a consequence, the proof of the instantiation Lemma 44 for the calculus of Table 1 extended with the tense operators does not work. In fact, this extended calculus *is not sound*: remember that it derives the Barcan formula and the necessity of the difference, which are easily seen to be invalid (already in the case of Kripke semantics). As we shall see in Table 2 of next subsection, the semantic failure of such logical principles is tightly related (in fact, it is equivalent to) the failure of the converse of $(C\Diamond_p)$.

We come across a very similar problem if we try to extend topological semantics beyond the etale spaces case [56], [51]. Fix a topological space S. A bundle over S (or, simply, a *bundle*, leaving S as understood) is a topological space D endowed with a continuous map $d : D \longrightarrow S$; a map among bundles $d : D \longrightarrow S$ and $d' : D' \longrightarrow S$ is a continuous map $f : D \longrightarrow D'$ such that $d' \circ f = d$. Let $\mathbf{Top}/S$ be the category of bundles and related maps; this category has products (which are fibered products over S). Moreover, for every bundle D, the powerset $\mathcal{P}(D)$ of D carries an interior algebra structure. Thus, we can come back to the hyperdoctrinal point of view and use Definitions 41, 42, 43 to introduce bundle models. Equivalently, we may replace Definition 43 by the corresponding forcing conditions: for instance, (T) is the forcing condition for $\Box$.[31]

Once again, however, the trouble with bundles is the instantiation Lemma 44. In fact, the notion of a modal hyperdoctrine should be modified [55], [56][32] in order to adapt it to bundles: the powerset functor fails to give a modal hyperdoctrine over $\mathbf{Top}/S$ (in the sense of Definition 40), because if $f : D \longrightarrow D'$ is a continuous function, the inverse image function $\mathcal{P}(f) := f^{-1} : \mathcal{P}(D)' \longrightarrow \mathcal{P}(D)$ is not an interior algebras morphism, we only have

$$(C\Box) \qquad\qquad f^{-1}\Box(A) \subseteq \Box f^{-1}A$$

for every $A \subseteq D'$ (here $\Box$ obviously denotes the topological interior operator). Notice that condition $(C\Box)$ is very similar to condition $(C\Diamond_p)$: the former expresses the continuity definition with respect to interior, whereas the latter expresses 'continuity with respect to a kind of closure'. The calculus of Table 1 is not sound with respect to bundle models:[33] there is no surprise in that, *such calculus wrongly assumes* (through Definition 39(d)) *the converse of* $(C\Box)$ *to hold, i.e., it assumes that all continuous functions are open.* Notice that, on the contrary, continuous maps among etale spaces *are* open: this is why we did not meet any trouble in Example 5 from Subsection 10.1.

In next section, we shall get an axiomatization of bundle models. Since, when all involved topologies are preordered, bundle models reduce to counterpart models and since the latter are sufficient for completeness, we prefer to directly skip to counterpart models.

11 COUNTERPART SEMANTICS

The previous subsection's analysis of the interplay between substitution and modal operators in mathematically motivated models (topological bundles, essential geometric morphisms of toposes) suggests the revision of some basic syntactic definitions from Subsection 9.1: in this way, a *continuity condition* can be easily formulated. Continuity turns out to be *the only axiom schema needed in order to axiomatize counterpart semantics.* We shall devote the present final section to the illustration of this (surprisingly) very simple axiomatization taken from [55].

[31] Bundle models should not be confused with the models over Kripke bundles of [111] (such Kripke bundles define a semantics which is intermediate between Kripke frames with equality and presheaf semantics [121]).

[32] The modification is simple: in Definition 40, just replace **Int** by the category having as objects interior algebras and as arrows the Boolean morphisms f such that $f(\Box a) \leq \Box f(a)$.

[33] For instance, the calculus of Table 1 derives the necessity of identity, which holds (when S is the singleton space) precisely for trivial discrete spaces.

Before philosophically motivating, introducing and discussing counterpart semantics, we prefer to immediately make the needed modifications at the syntactic level. In order to express the continuity axiom, we need to treat substitution carefully: in fact, the statement of the continuity axiom is that only one half of Definition 39(d) holds, the other half being an openness requirement which is not valid (the definition of a continuous map in topology says that inverse image of the interior is *contained* in the interior of inverse image). To get the desired effect, we treat modalized formulas as atomic formulas and change Definition 38(iv) as follows:

(iv') if $\phi : k$ is a k-formula and if $t_1 : n, \ldots, t_k : n$ are n-terms, then $(\Box\phi)(t_1, \ldots, t_k) : n$ is an n-formula.

We shall abbreviate $(\Box\phi)(x_1, \ldots, x_n) : n$ by $\Box\phi : n$. Next, we modify Definition 39(d) by:

(d') if $\underline{v} : m$ is an n-tuple of m-terms and $(\Box\phi)(t_1, \ldots, t_k) : n$ is an n-formula, we let the m-formula $(\Box\phi)(t_1, \ldots, t_k)[\underline{v}] : m$ be equal to $(\Box\phi)(t_1[\underline{v}], \ldots, t_k[\underline{v}]) : m$.

The 'counterpart' modal calculus **CS4** has again the rules and the axiom schemata of Table 1 (to be interpreted according to the new definition of formula and substitution) and, in addition, the following *continuity* axiom schemata:

$$(Cont) \qquad (\Box\phi)[t_1, \ldots, t_k] \to \Box(\phi[t_1, \ldots, t_k]) : n$$

(notice that $(\Box\phi)[t_1, \ldots, t_k]$ is precisely $(\Box\phi)(t_1, \ldots, t_k)$ by the above definitions and conventions).

11.1 Counterpart Models

The origins of counterpart semantics come from philosophical considerations: counterpart semantics is a way of overcoming transworld identification problems when evaluating *de re* statements in possible worlds semantics:

> The counterpart relation is our substitute for identity between things in different worlds. Where some would say that you are in several worlds, in which you have somewhat different properties and somewhat different things happen to you, I prefer to say that you are in the actual world and no other, but you have counterparts in several other worlds ([79], p. 114).

A large philosophical debate (see for instance [64], [42]) has pointed out both the merits and the difficulties of the counterpart-theoretic point of view in the semantics of quantified modal logic. We shall mainly concentrate here on *axiomatization* issues: these turn out to be useful for clarifying aspects of the counterpart doctrine that are commonly considered rather obscure.

Counterpart theory is introduced in [79] by axiomatizing the intuition behind the intended notion of a counterpart. Such intuition is provided by the concept of similarity: informally, a (living in w) is a counterpart of b (living in v) iff 'a resembles to b more closely than anything else living in w'. For the axiomatization of the notion of a counterpart, a suitable first-order theory I is introduced ([79], Section I). A translation of first-order modal language into the extensional language of I is then considered ([79], Section II):

such translation gives implicitly — as happens with standard translation in the Kripkean case — the inductive forcing conditions for counterpart semantics.[34] A counterpart relation is finally joined to a customary accessibility relation (in order to model systems whose propositional basis is not **S5**) and the translation from modal language into the counterpart extensional language is modified accordingly ([79], Section V).

The axioms of I express rather light constraints on the relation 'being a counterpart of': this is taken to be an almost generic binary relation among individuals. In particular, the counterpart relation needs not be symmetric nor transitive, individuals may have one, many, or no counterparts at all in any other world; however individuals live in a unique assigned world and are their own unique counterpart in the world they live in ([79], p.114).

Models of I can consequently be represented as triples $d : D \longrightarrow W$, where W is the set of possible worlds, D is the set of individuals and the function d specifies the world $d(a)$ a given individual $a \in D$ lives in. Both W and D are endowed with binary relations: these are the counterpart relation R_D for D and the accessibility relation R_W for W. The function d must preserve such relation: this means that, whenever b is counterpart of a, then the world $d(b)$ where b lives in should be accessible from the world $d(a)$ where a lives in. Formally we have that

(8) $\quad aR_Db \;\Rightarrow\; d(a)R_Wd(b)$

holds for all $a, b \in D$. Although Lewis does not make assumption (8) in Section V of [79], this assumption is implicit in the translation instructions he gives from the modal language to the extensional language.[35] We shall call *Lewis triples* the triples $d : (D, R_D) \longrightarrow (W, R_W)$ satisfying (8).

Since we want to make a close comparison with the topological bundles of Subsection 10.2, we assume that both D and W are preodered sets: this means that both the accessibility relation R_W and the counterpart relation R_D are assumed to be reflexive and transitive. We point out that we do that *just for enlightening comparisons with topological bundle semantics to be immediate.* In fact, dropping reflexivity and transitivity for both the counterpart and the accessibility relation, would simply result in treating **K**-based systems: this is quite easily done in our framework, because once **S4** axioms are removed from the syntax and reflexivity-transitivity requirements are removed from the semantics, *soundness and completeness results of Subsection 11.2 extend trivially.*

Lewis' requirement that every individual a living in a world w has itself as counterpart in w (together with reflexivity of the accessibility relation) is precisely what is needed in order to make axiom **T** valid: for this reason, this requirement is included in our setting, but should be dropped for **K**-based systems. Finally, Lewis requirement that a is the only counterpart of a in the world a lives in, is a non-modally definable condition which does not affect soundness and completeness of our systems (as will be evident from the completeness proof of Subsection 11.2).

[34] Other translations (different from Lewis' original one) have been proposed by G. Forbes and M. Ramachandran: see the recent paper [28] for an essential account of them and for the relevant pointers to the literature.

[35] In other words, the translation of a modal sentence is not influenced by this extra assumption: quantifiers are relativized to individuals living in the current world w and, when the translation of a modalized formula requires taking into consideration another world v, then v must be accessible from w and the translation instruction takes care of replacing current free variables ranging over individuals living in w by variables ranging over counterparts of them living all in v (see (T2i*)-(T2j*) in [79], p. 125).

Thus we restrict our considerations to reflexive-transitive Lewis triples: these are the triples $d : (D, \leq) \longrightarrow (W, \leq)$, where $(W, \leq)$ and $(D, \leq)$ are preordered sets and d is an order-preserving map. *This is nothing but a special case of the notion of a topological bundle from Subsection 10.2*: more precisely, it is the special case arising when topologies are preordered.[36]

Given a reflexive-transitive Lewis triple $d : (D, \leq) \longrightarrow (W, \leq)$ and a world $w \in W$, we can associate with it the fiber over w, namely the set $D_w = \{a \in D \mid d(a) = w\}$; whenever $v \leq w$, we can also consider the relation $D_{vw} = \{(a, b) \mid a \in D_v, b \in D_w$ and $a \leq b\}$, obtained from the restriction of the counterpart relation to $D_v \times D_w$. Now the collection of sets $\{D_w \mid w \in W\}$ and the collection of relations $\{D_{vw} \mid v \leq w\}$ form a $(W, \leq)$-relational domain, in the sense of the definition below. Relational domains and reflexive-transitive Lewis triples are equivalent formalisms (there are precise and easy technical results guaranteeing that),[37] so that we prefer to move to the relational domain formalism.

We have finally arrived at the relevant formal definitions from [55], [56], [51]. Given a preordered set $\mathfrak{F} = (W, \leq)$, by an *$\mathfrak{F}$-relational domain* D, we mean a collection of sets $\{D_w \mid w \in W\}$, endowed for every pair $v \leq w \in W$ with a relation $D_{vw} \subseteq D_v \times D_w$. Such 'transition' relations are assumed to satisfy the following requirements for all $v \leq w \leq z \in W$, for all $a \in D_v, b \in D_w, c \in D_z$:

$$(L) \qquad\qquad aDa, \qquad aDb \;\&\; bDc \;\Rightarrow aDc$$

(where we write aDb for $(a, b) \in D_{vw}$, etc.). Whenever aDb holds (for $v \leq w, a \in D_v, b \in D_w$) we say that b is a *w-counterpart* (or simply a counterpart) of a.

The following list of further definitions/notational conventions will complete our counterpart domain settings. Let an $\mathfrak{F}$-relational domain D be given; then given tuples $\underline{a} = (a_1, \ldots, a_n) \in D_v^n$, $\underline{b} = (b_1, \ldots, b_n) \in D_w^n$ (with $v \leq w$), the notation $\underline{a}D\underline{b}$ means that a_iDb_i holds for all $i = 1, \ldots, n$. We also use the notation $\underline{a}_i$ to mean the i-th component of the tuple $\underline{a}$ (i.e., if $\underline{a} = (a_1, \ldots, a_n)$, then $\underline{a}_i$ stands for a_i).

A *$\mathfrak{F}$-relational map* $f : D \longrightarrow D'$ among $\mathfrak{F}$-relational domains is a collection of functions $\{f_v : D_v \longrightarrow D'_v \mid v \in W\}$ satisfying the requirement $aDb \;\Rightarrow f_v(a)D'f_w(b)$ (for every $v \leq w, a \in D_v, b \in D_w$). The n-th cartesian power of an $\mathfrak{F}$-relational domain D is the relational domain D^n so specified: $(D^n)_w = (D_w)^n$ and $\underline{a}D^n\underline{b}$ holds (for $v \leq w, \underline{a} \in D_v^n, \underline{b} \in D_w^n$) iff $\underline{a}_iD\underline{b}_i$ holds for every i.[38] For $n = 0$, the singleton $\mathfrak{F}$-relational domain D^0 is so described: D_w^0 contains for every w just the empty tuple $*$ from D_w and we have $*D^0*$ for all $v \leq w \in W$. A sub-relational domain $S \subseteq D$ of an $\mathfrak{F}$-relational domain D is a collection of subsets $S = \{S_w \subseteq D_w \mid w \in W\}$. The following pair of definitions is the expected one:

---

[36] Recall that a topology is said to be preordered iff there is a preorder relation such that the open sets coincide with the upward closed subsets. Notice also that a topology is preordered iff open sets are closed under arbitrary (not just finite) intersections.

[37] We mention precise connections for the interested reader: let **Preord** be the category of preordered sets and let $\mathfrak{F}$ be a preorder. Then the category of counterpart frames $\mathbf{Rel}^{\mathfrak{F}}$ over $\mathfrak{F}$ is equivalent [56], [89] to the category of reflexive-transitive Lewis triples over $\mathfrak{F}$, which is nothing but the slice category $\mathbf{Preord}/\mathfrak{F}$; the latter is embedded in $\mathbf{Top}/\mathfrak{F}$ (both the equivalence and the embedding extend to the modal hyperdoctrinal level).

[38] Thus D^n is the n-th cartesian power of D in the category $\mathbf{Rel}^{\mathfrak{F}}$ [55],[56],[89] of $\mathfrak{F}$-relational domains and $\mathfrak{F}$-relational maps. Similarly, sub-relational domains defined below correspond to regular subobjects in that category.

DEFINITION 50. Let $\mathcal{L}$ be a language and let $\mathfrak{F}$ be a frame; an $\mathfrak{F}$-counterpart model (or, simply an $\mathfrak{F}$-model) $\mathfrak{M}$ consists of the following data:

- an $\mathfrak{F}$-relational domain D (the domain of $\mathfrak{M}$);
- an interpretation $\mathcal{I}$ mapping every function symbol f to an $\mathfrak{F}$-relational map $\mathcal{I}(f) : D^n \longrightarrow D$ and every predicate symbol P to a sub-relational domain $\mathcal{I}(P) \subseteq D^n$ (here n is clearly the arity of f, P).

It is assumed that $\mathcal{I}(\bot) \subseteq D^0$ is the empty sub-relational domain and that $\mathcal{I}(=)_w = \{(a, a) \mid a \in D_w\}$.

DEFINITION 51. Given an $\mathfrak{F}$-model $\mathfrak{M} = (D, \mathcal{I})$ and an n-term $t : n$, the $\mathfrak{F}$-relational map $[\![t]\!]^{\mathfrak{M}} : D^n \longrightarrow D$ is so defined by induction (for all $w \in W$, $\underline{a} \in D^n_w$):

- if $t = x_i$ $(i = 1, \ldots, n)$, then $[\![t]\!]^{\mathfrak{M}}_w(\underline{a}) := \underline{a}_i$;
- if $t = f(t_1, \ldots, t_k)$, then $[\![t]\!]^{\mathfrak{M}}_w(\underline{a}) := \mathcal{I}(f)_w([\![t_1]\!]^{\mathfrak{M}}_w(\underline{a}), \ldots, [\![t_k]\!]^{\mathfrak{M}}_w(\underline{a}))$.

We must now define forcing conditions: this is done in complete analogy with Lewis translation instructions (T2i)-(T2j) from [79] (or, better, as we also have an accessibility relation, with Lewis translation instructions (T2i*)-(T2j*) from Section V of [79]). Notice that: (i) we use model-theoretic forcing instead of a pure syntactic translation into the extensional language of I; (ii) we have typed languages, so our model-theoretic forcing uses finitary assignments; (iii) we modified above the standard definition of a formula, so our truth clause for formulas whose main connective is a modal operator is a combination of the truth clause for $\Box$ and of the truth clause for atomic formulas.

DEFINITION 52. Given an $\mathfrak{F}$-model $\mathfrak{M} = (D, \mathcal{I})$, an n-formula $\phi : n$, a world $w \in W$ and a tuple $\underline{a} \in D^n_w$, the forcing relation $\underline{a} \models^{\mathfrak{M}}_w \phi$ is so defined, by induction:

$$
\begin{array}{lll}
\underline{a} \models^{\mathfrak{M}}_w P(t_1, \ldots, t_k) & \text{iff} & ([\![t_1]\!]^{\mathfrak{M}}_w(\underline{a}), \ldots, [\![t_k]\!]^{\mathfrak{M}}_w(\underline{a})) \in \mathcal{I}(P)_w; \\
\underline{a} \models^{\mathfrak{M}}_w \psi_1 \to \psi_2 & \text{iff} & (\underline{a} \models^{\mathfrak{M}}_w \psi_1 \ \Rightarrow\ \underline{a} \models^{\mathfrak{M}}_w \psi_2); \\
\underline{a} \models^{\mathfrak{M}}_w \forall x_{n+1}\psi & \text{iff} & \text{for all } a \in D_w,\ (\underline{a}, a) \models^{\mathfrak{M}}_w \psi; \\
\underline{a} \models^{\mathfrak{M}}_w (\Box\psi)(t_1, \ldots, t_k) & \text{iff} & \text{for all } w \le v,\ \underline{b} \in D^k_v \\
 & & ([\![t_1]\!]^{\mathfrak{M}}_w(\underline{a}), \ldots, [\![t_k]\!]^{\mathfrak{M}}_w(\underline{a})) D \underline{b} \ \Rightarrow\ \underline{b} \models^{\mathfrak{M}}_v \psi.
\end{array}
$$

We say that $\phi : n$ is valid in $\mathfrak{M}$ iff $\underline{a} \models^{\mathfrak{M}}_w \phi$ holds for all $w \in W, \underline{a} \in D^n_w$.

We underline that, in case $\phi : n$ is just $\Box\psi : n$, the above forcing condition says that *the tuple $\underline{a}$ satisfies $\Box\psi : n$ in w iff for all $w \le v$, all tuples made of v-counterparts of the $\underline{a}$'s satisfy $\psi : n$ in v.* This is just Lewis' translation instruction written in our notation; moreover it coincides with the topological forcing condition (T) from Section 10, in the case of topologies induced by a preorder relation.

11.2 Soundness and Completeness

For the soundness theorem, we first need a suitable instantiation lemma, which is easily proved by induction:

LEMMA 53. *For all $\phi : k$, $v : k$, $t_1 : n, \ldots, t_k : n$ and for all $\underline{a} \in D^n_w$, we have in any $\mathfrak{F}$-model $\mathfrak{M} = (D, \mathcal{I})$:*

(i) $[\![v[t_1,\ldots,t_k]]\!]_w^{\mathfrak{M}}(\underline{a}) \;=\; [\![v]\!]_w^{\mathfrak{M}}([\![t_1]\!]_w^{\mathfrak{M}}(\underline{a}),\ldots,[\![t_k]\!]_w^{\mathfrak{M}}(\underline{a}))$;

(ii) $\underline{a} \models_w^{\mathfrak{M}} \phi[t_1,\ldots,t_k]$ iff $([\![t_1]\!]_w^{\mathfrak{M}}(\underline{a}),\ldots,[\![t_k]\!]_w^{\mathfrak{M}}(\underline{a})) \models_w^{\mathfrak{M}} \phi$.

THEOREM 54. *(Validity). If $\phi : n$ is provable in* **CS4***, then it is valid in every $\mathfrak{F}$-model $\mathfrak{M}$.*

Proof. Rule (*Inst*) is valid by Lemma 53, whereas **S4**-propositional tautologies are valid by conditions (L). Validity of (MP), (Nec), ($\forall$-Ex), ($\forall$-In), ($Refl$) and ($Repl$) are trivial: using the instantiation Lemma 53, we have e.g. for the case of ($\forall$-Ex) (let $\underline{a} = (a_1,\ldots,a_{n+1})$), $\underline{a} \models_w^{\mathfrak{M}} (\forall x_{n+1}\,\psi)[x_1,\ldots,x_n]$ iff $([\![x_1]\!]_w^{\mathfrak{M}}(\underline{a}),\ldots,[\![x_n]\!]_w^{\mathfrak{M}}(\underline{a})) \models_w^{\mathfrak{M}} \forall x_{n+1}\,\psi$ iff $(a_1,\ldots,a_n) \models_w^{\mathfrak{M}} \forall x_{n+1}\,\psi$ and the latter implies $(a_1,\ldots,a_{n+1}) \models_w^{\mathfrak{M}} \psi$, that is $\underline{a} \models_w^{\mathfrak{M}} \psi$. Validity of the continuity schema ($Cont$) follows from the definition of an $\mathfrak{F}$-relational map and Lemma 53 again. ❑

We now sketch in some detail the completeness proof. The first-order (non-modal) language $\mathcal{L}_c$ is obtained by adding to $\mathcal{L}$ a new n-ary predicate symbol $P_{\Box\phi}$ for every n-formula of the kind $\Box\phi : n$ of $\mathcal{L}$. To every modal formula $\psi : m$ of $\mathcal{L}$ we assign a non-modal formula $\psi_c : m$ of $\mathcal{L}_c$ simply by replacing the subformulas of the kind $(\Box\phi)(t_1,\ldots,t_k) : m$ by the atomic formulas $P_{\Box\phi}(t_1,\ldots,t_k) : m$. The classical first-order theory T_c has as proper axioms the formulas $\phi_c : n$ such that $\phi : n$ is provable in **CS4**. Notice that all formulas of $\mathcal{L}_c$ are of the kind $\psi_c : m$ (for suitable $\phi : m$) and that $\psi : m$ turns out to be provable in **CS4** iff $\psi_c : m$ is provable in the classical first-order theory T_c (one direction is trivial from the construction of T_c, for the other side simply observe that T_c has no deductive machinery which is not already available in **CS4**).[39]

We use the letters $w, v, \ldots$ to denote T_c-models (i.e., classical first-order $\mathcal{L}_c$-structures making the universal closures of the proper axioms of T_c true); for a T_c-model w, we use the notation D_w to denote the domain of w, $[\![t]\!]_w$ to denote the interpretation $D_w^n \longrightarrow D_w$ of the term $t : n$ in w, $\underline{a} \models_w \phi_c$ to denote the (finitary assignments) forcing relation in w with respect to classical first-order Tarski semantics.

An *admissible relation* R among T_c-models w, v is a relation $R \subseteq D_w \times D_v$ satisfying the following two requirements, for every $n \geq 0$ and for every $\underline{a} \in D_w^n, \underline{b} \in D_v^n$:

(A1) if $\underline{a}_i R \underline{b}_i$ holds for every $i = 1,\ldots,n$, then $[\![t]\!]_w(\underline{a})\ R\ [\![t]\!]_v(\underline{b})$ holds for every n-term $t : n$;

(A2) if $\underline{a}_i R \underline{b}_i$ holds for every $i = 1,\ldots,n$, then

$$\underline{a} \models_w P_{\Box\phi}(x_1,\ldots,x_n) \;\Rightarrow\; \underline{b} \models_v \phi_c$$

holds for every n-formula $\phi : n$.

Notice that, thanks to the propositional **S4**-axioms, identity relations are admissible and admissible relations compose.

Following model-theoretic terminology, we call *n-type* a set Γ of n-formulas of $\mathcal{L}_c$; an n-type is T_c-consistent iff for no finite conjunction $\psi_c : n$ of formulas from Γ, the formula $\neg\psi_c : n$ is provable in T_c. For a T_c-model w and for an n-tuple $\underline{a} \in D_w^n$, the n-type $\Gamma_{\underline{a}}$ is

[39] For classical first-order logic, we keep for uniformity reasons the same typed framework of Subsection 9.1.

the set of n-formulas $\psi_c : n$ such that $\underline{a} \models_w \psi_c$. An n-type Γ is *realized* in w iff for some $\underline{a} \in D_w^n$ we have $\Gamma \subseteq \Gamma_{\underline{a}}$. It follows from completeness theorem for first-order theories, that a T_c-consistent n-type is always realized in a T_c-model. The following lemma is a crucial step:

LEMMA 55. *Let w, v be T_c-models, $\underline{a} \in D_w^n, \underline{b} \in D_v^n$; if $\{\phi_c : n \mid \underline{a} \models_w P_{\Box\phi}(x_1, \ldots, x_n)\} \subseteq \Gamma_{\underline{b}}$, then there is an admissible relation $D_{wv} \subseteq D_w \times D_v$ such that $\underline{a}_i D_{wv} \underline{b}_i$ holds (for every $i = 1, \ldots, n$).*

Proof. Define D_{wv} as follows: say that $aD_{wv}b$ holds iff there is a term $t : n$ such that $[\![t]\!]_w(\underline{a}) = a$ and $[\![t]\!]_v(\underline{b}) = b$. Condition (A1) holds trivially, whereas for condition (A2) we need the continuity axiom schema. In fact, fix a k-tuple $t_1 : n, \ldots, t_k : n$ of n-terms such that

$$([\![t_1]\!]_w(\underline{a}), \ldots, [\![t_k]\!]_w(\underline{a})) \models_w P_{\Box\phi}(x_1, \ldots, x_k)$$

holds; by the instantiation Lemma 53(ii) (which holds in classical first-order semantics as well), we get $\underline{a} \models_w P_{\Box\phi}(t_1, \ldots, t_k)$. Since $(\Box\phi)[t_1, \ldots, t_k] \to \Box(\phi[t_1, \ldots, t_k]) : n$ is provable in **CS4**, by the definition of T_c we have $\underline{a} \models_w P_{\Box(\phi[t_1,\ldots,t_k])}(x_1, \ldots, x_n)$. By the hypothesis of the lemma, we get $\underline{b} \models_v (\phi[t_1, \ldots, t_k])_c = \phi_c[t_1, \ldots, t_k]$, that is (by the instantiation lemma) $([\![t_1]\!]_v(\underline{b}), \ldots, [\![t_k]\!]_v(\underline{b})) \models_v \phi_c$. ❑

THEOREM 56. *(Completeness).*[55], [51] *If a formula $\phi : n$ is not provable in* **CS4**, *then there are a frame $\mathfrak{F}$ and an $\mathfrak{F}$-model $\mathfrak{M} = (D, \mathcal{I})$, in which $\phi : n$ is not valid.*

Proof. We use the 'subordination frame' technique [67][40] and inductively define a tree $\mathfrak{F} = (W, \leq)$ of T_c-models and of admissible relations as follows. The root of $\mathfrak{F}$ is any T_c-model realizing the T_c-consistent n-type $\{\neg\phi_c : n\}$. Let the node w of $\mathfrak{F}$ be already defined; for every pair $\langle \underline{a}, \Box\psi : m\rangle$ such that $\underline{a} \not\models_w P_{\Box\psi}(x_1, \ldots, x_m)$, we take a T_c-model v, an admissible relation $D_{wv} \subseteq D_w \times D_v$ and a tuple $\underline{b}$ from D_v such that $\underline{b} \not\models_v \psi_c$ and $\underline{a}_i D_{wv} \underline{b}_i$ (for $i = 1, \ldots, n$): this is possible, by Lemma 55, because the m-type

$$\{\psi'_c : m \mid \underline{a} \models_w P_{\Box\psi'_c}(x_1, \ldots, x_m)\} \cup \{\neg\psi_c : m\}$$

is T_c-consistent by propositional modal validities. These v and these D_{wv} are all added to the already defined part of $\mathfrak{F} = (W, \leq)$ in the inductive step of the construction.

Now that $\mathfrak{F}$ has been built, it is clear that the collection of the supports D_w — and of the compositions of the admissible relations D_{wv} introduced during the construction of $\mathfrak{F}$ — gives in the obvious way an $\mathfrak{F}$-relational domain D. Similarly, the collection of the interpretation functions of the various T_c-models $w \in W$ can be glued together to form a global interpretation function $\mathcal{I}$ (notice, however, that (A1) is needed here to show that the global interpretation of a function symbol is an $\mathfrak{F}$-relational map). It remains to prove a standard 'truth-lemma', namely that we have

$$\underline{a} \models_w \psi_c \quad \text{iff} \quad \underline{a} \models_w^{\mathfrak{M}} \psi$$

for all $\psi : m, w \in W, \underline{a} \in D_w^m$. This is trivial, however, by the construction of $\mathfrak{F}$ and by (A2). ❑

[40]If we allow categories (and not just preordered sets) to be frames, there is an obvious 'canonical-model'-like technique that works: just take as $\mathfrak{F}$ the category having T_c-models as objects and admissible relations as arrows.

As pointed out above, Theorem 56 covers also bundle semantics.[41] A past-possibility tense operator $\Diamond_p$ can be added both to the syntax and to the semantics (Theorems 54 and 56 extend trivially): we call the related system $\mathbf{CS4_t}$. More precisely, to get $\mathbf{CS4_t}$, we add for instance to Table 1 the axiom schema $\phi \to \Box\Diamond_p\psi : n$ and the rule $\phi \to \Box\psi : n / \Diamond_p\phi \to \psi : n$ (notice that the continuity principle $\Diamond_p(\phi[\underline{t}]) \to (\Diamond_p\phi)[\underline{t}] : n$ for $\Diamond_p$ follows from the corresponding continuity principle (*Cont*) for $\Box$). The semantic forcing condition for $\Diamond_p$ says that a n-tuple $\underline{a}$ from D_w^n forces $(\Diamond\psi)(t_1, \ldots, t_k) : n$ iff there is a k-tuple $\underline{b}$ that forces $\psi : k$ *of which* $([\![t_1]\!]_w^{\mathfrak{M}}(\underline{a}), \ldots, [\![t_k]\!]_w^{\mathfrak{M}}(\underline{a}))$ is a k-tuple of (respective) w-counterparts.

Natural conditions (like being totally defined, partial functions, etc.) can be imposed on the transition relations D_{vw}: such conditions are modally axiomatizable by suitable axiom schemata in $\mathbf{CS4_t}$. The summary of such extensions [55] of Theorems 54 and 56 is given in Table 2 below:[42] the meaning of that Table is that the semantic condition in the third column is axiomatized by any one of the equivalent axiom schemata written in the corresponding second column (the notation D_{vw}^o means the converse relation of D_{vw}). Notice that the equivalent axiom schemata in the second column sometimes involve properties of substitutions and sometimes they are just well-known standard modal principles (this explains why, if properties of substitutions are built-in in the language through definitions like 39(d), then *there is no way of blocking the derivation of the corresponding modal principles*).

Table 2.

No.	Equivalent Axiom Schemata	Semantic Conditions
1	$z_1 = z_2 \to \Box(z_1 = z_2) : 2$ $\Diamond_p(z_1 = z_2) \to z_1 = z_2 : 2$ $\Box(\phi[\underline{x}, y, y]) \to (\Box\phi)[\underline{x}, y, y] : n{+}1$	all D_{vw} are partial functions
2	$\exists y\Box\phi \to \Box\exists y\phi : n$ $\Diamond_p\exists y\phi \to \exists y\Diamond_p\phi : n$ $\Box(\phi[\underline{x}]) \to (\Box\phi)[\underline{x}] : n{+}1$	all D_{vw} are totally defined relations
3	$z_1 \neq z_2 \to \Box(z_1 \neq z_2) : 2$ $\Diamond_p(z_1 \neq z_2) \to z_1 \neq z_2 : 2$ $(\Diamond_p\phi)[\underline{x}, y, y] \to \Diamond_p(\phi[\underline{x}, y, y]) : n{+}1$	all D_{vw}^o are partial functions
4	$\forall y\Box\phi \to \Box\forall y\phi : n$ $\Diamond_p\forall y\phi \to \forall y\Diamond_p\phi : n$ $(\Diamond_p\phi)[\underline{x}] \to \Diamond_p(\phi[\underline{x}]) : n{+}1$	all D_{vw}^o are totally defined relations

11.3 Conclusions

At the end of a lengthy detour around alternative semantics for predicate modal logics, we met in the present section systems $\mathbf{CS4}$, $\mathbf{CS4_t}$ together with the 'old-style' alternative

[41] For plain topological semantics (i.e., for bundles over the singleton topological space), an extra axiom schema is needed, see [56], [51].

[42] In particular, the axiomatization of the essential geometric morphisms semantics outlined at the beginning of Subsection 10.2 corresponds to lines 1+2 of Table 2. From the axiom schemata of lines 1+2, the openness principle $(\Box\phi)[\underline{t}] \leftrightarrow \Box(\phi[\underline{t}]) : n$ can be derived for $\Box$, but *not* for $\Diamond_p$.

semantics proposed in the sixties by D. Lewis. In fact, although systems **CS4**, $\mathbf{CS4_t}$ were built in [55], [56], [51] in order to axiomatize the mathematical frameworks outlined in Subsection 10.2, we just showed that such systems axiomatize counterpart semantics as well. We feel that this unexpected confluence among mathematically and philosophically motivated research should be taken seriously. We conclude the second part of the chapter by giving some more information on this topic.

- In Definition 52, variables are given only *local* values: this is because Definition 52 uses finitary assignments relative only to the possible world in which a formula is evaluated. This seems to be in accordance with concrete use of free variables as indexicals in linguistics [19] (reference of indexicals is fixed by the pragmatic context in which a sentence containing them is uttered). Moreover, *de re* statements like 'you will eventually spend holidays at the seaside' or 'some actual women will be sooner or later elected as the president of the USA' are correctly evaluated by referring to a future state of affairs in which the individual in question *exists*.[43] This is why, as already pointed out in [79] (p. 124), the converse of the Barcan formula is a *valid* modal principle: if $\exists y \Diamond P(y)$ is true at w, then there is $a \in D_w$ which has a v-counterpart a' enjoying P and this implies that $\Diamond \exists y P(y)$ is true in w as well. The role played by the converse of the Barcan formula in varying domains semantics is played here by the axiom schemata of line 2 of Table 2: such axiom schemata are valid whenever individuals never 'die' in accessible worlds (from the topological point of view, the axiom schemata of line 2 of Table 2 express the special condition that projections from fibered products are open maps).

- Our axiomatization of counterpart semantics seems to clarify some the classical objections that have been raised against it. For instance, Kripke argued in [74] that since the necessity of identity

  (9) $x_1 = x_2 \to \Box(x_1 = x_2) : 2$

  (which is invalid in counterpart semantics) follows from

  (10) $\Box(x_1 = x_1) : 1$

  and from

  (11) $x_1 = x_2 \wedge \Box(x_1 = x_1) \to \Box(x_1 = x_2) : 2$

  then the counterpart theorist is either forced to reject the Leibniz principle of replacement or to reject the evident fact that everything is necessarily equal to itself. However, (11) is *not* an example of the Leibniz principle: the latter principle (which is obtained from the axiom schema (*Repl*) and the rule (*Inst*) of Table 1) can be formulated as:

  (12) $u = v \wedge \phi[\underline{x}, u] \to \phi[\underline{x}, v] : n$

  (where $\phi : n+1$ is an $n+1$-formula and u, v are n-terms). That (11) cannot be confused with the Leibniz principle is clear from the following counterexample

[43] Notice that no existence predicate is used in Lewis' counterpart semantics (but see [23] for an extension in this sense).

(which is a special case of topological bundle semantics from Subsection 10.2): interpret n-ary predicate letters as subsets of $\mathbb{R}^n$, let identity, quantifiers, and Boolean connectives be interpreted as in first-order extensional Tarski semantics and let $\Box$ be interpreted as the interior operator in the product Euclidean topology. It is clear that in this interpretation any correct version (like (12)) of the Leibniz principle holds (trivially, equal points belong to the same subsets). Moreover, the interior of $x_1 = x_1$ contains all points (because the interior of the whole space is the whole space), which means that (10) is valid. Nevertheless, the necessity of the identity (9) is false, because the interior of the diagonal in the plane is empty. Consequently, we must admit that in order to 'derive' (9) we used something quite different from the Leibniz principle (and, as a matter of fact, we used (11) which is easily seen to be invalid).

- Notice that the formula $x_1 = x_2 \wedge \Box R(x_1, x_2) \to \Box R(x_1, x_1) : 2$ is valid, whereas the formula $x_1 = x_2 \wedge \Diamond R(x_1, x_2) \to \Diamond R(x_1, x_1) : 2$ is not. The reason why this happens is that the continuity principle is different for 'interior' and 'closure' (to derive the former formula we need $(\Box R(x_1, x_2))[x_1, x_1] \to \Box(R(x_1, x_2)[x_1, x_1]) : 2$ which is not available for $\Diamond$, because the continuity principle for $\Diamond$ is the converse). This explains why, in order to axiomatize counterpart semantics, we cannot simply use 'generic restrictions' to quantifiers and identity principles that apply to arbitrary intensional contexts.

- In our typed systems, there is a very natural way of formalizing the *de re*/*de dicto* distinctions: we can use $(\Box\phi)[\underline{t}] : n$ and $\Box(\phi[\underline{t}]) : n$, respectively. Constants which are non-rigid designators[44] can consequently be treated either by dropping the continuity principle for them or, in a more uniform way, by taking a suitable many-sorted approach [54] (validity and completeness theorems extend trivially). The notation $(\Box\phi)[\underline{t}] : n$ is very similar to λ-abstraction notation of intensional semantics: in **CS4**-like systems, it can be seen just as a notational variant of λ-abstraction whenever the axiom schema $\Box(\phi[\underline{x}]) \to (\Box\phi)[\underline{x}] : n{+}1$ from line 2 of Table 2 holds.

- Whenever the axiom schemata from line 2 of Table 2 do not hold however (i.e., whenever transition relations are not totally defined), the λ-notation is insufficient if types are omitted: this is because $(a_1, a_2) \models_w^{\mathfrak{M}} \Diamond P(x_1)$ may be false when $a_1 \models_w^{\mathfrak{M}} \Diamond P(x_1)$ holds (a_2 may not have enough counterparts). Hence, the formulas $\Diamond P(x_1) : 2$ and $\Diamond P(x_1) : 1$ should be kept as distinct: the former is relative to contexts in which two free variables (say, two indexicals) have been assigned a value, the latter to contexts in which only one variable has been assigned a value.

- The need for types explains why sometimes people got the wrong impression that basic principles like Aristotle's law fail in Lewis' semantics. What may fail is a formula like $\Box(P(x_1) \to Q(x_2)) \to ((\Box P(x_1))[x_1] \to (\Box Q(x_1))[x_2]) : 2$, which is *not* an example of the Aristotle schema; the latter obviously is $\Box(\phi \to \psi) \to (\Box\phi \to \Box\psi) : n$ and, as such, it can be instantiated only into valid formulas, like e.g. $\Box(P(x_1) \to Q(x_2)) \to (\Box P(x_1) \to \Box Q(x_2)) : 2$.

[44] A non-rigid designator constant c is interpreted as a collection of elements $\{c_w \in D_w \mid w \in W\}$ such that c_v needs not be a v-counterpart of c_w when $w \leq v$ holds. Using the isomorphism of $\mathfrak{F}$-relational domains and preordered bundles over $\mathfrak{F}$, these are exactly arbitrary *non-continuous* functions from the terminal bundle $\mathfrak{F}$ to D.

Additional comments can be found in [54]; still the overall implications of the above systematization of counterpart semantics need to be further explored.

ACKNOWLEDGEMENTS

The first author wishes to thank the participants of a research seminar at Roskilde University for comments on the first part of the chapter. The second author wishes to thank all the people (a partial list includes: S. Artemov, J. van Benthem, G. Corsi, C. Pizzi, V.B. Shehtman, D.P. Skvortsov, F. Wolter, M. Zakharyaschev), who helped him in collecting relevant important information.

BIBLIOGRAPHY

[1] H. Andréka, I. Nemeti, and J. VanBenthem. Modal languages and bounded fragments of predicate logics. *Journal of Philosophical Logic*, 27:217–274, 1998.

[2] C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation and complexity. *Journal of Symbolic Logic*, 66:977–1010, 2001.

[3] C. Areces, P. Blackburn, and M. Marx. Repairing the interpolation theorem in quantified modal logic. *Annals of Pure and Applied Logic*, 124:287–299, 2003.

[4] S. N. Artëmov and G. K. Dzhaparidze. Effective predicate logics of provability. *Dokl. Akad. Nauk SSSR*, 297(3):521–523, 1987.

[5] Sergei Artëmov and Giorgie Dzhaparidze. Finite Kripke models and predicate logics of provability. *J. Symbolic Logic*, 55(3):1090–1098, 1990.

[6] R. C. Barcan. A functional calculus of first order based on strict implication. *Journal of Symbolic Logic*, 11:1–16, 1946.

[7] E. Bencivenga. Free logics. In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, 2nd Edition*, volume 5, pages 147–196. Kluwer Academic Publishers, 2002.

[8] R. Berger. The undecidability of the domino problem. *Memoirs of the Amer. Math. Soc.*, 66, 1966.

[9] P. Blackburn. Internalizing labelled deduction. *Journal of Logic and Computation*, 10:137–168, 2000.

[10] P. Blackburn. Arthur Prior and hybrid logic. *Synthese*, 2006. Special issue edited by T. Braüner, P. Hasle, and P. Øhrstrøm.

[11] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.

[12] P. Blackburn and M. Marx. Tableaux for quantified hybrid logic. In U. Egly and C. Fermüller, editors, *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX 2002*, volume 2381 of *Lecture Notes in Artificial Intelligence*, pages 38–52. Springer-Verlag, 2002.

[13] P. Blackburn and M. Marx. Constructive interpolation in hybrid logic. *Journal of Symbolic Logic*, 68:463–480, 2003.

[14] Francis Borceux. *Handbook of categorical algebra*, volume 50-52 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1994. Basic category theory.

[15] Egon Börger, Erich Grädel, and Yuri Gurevich. *The classical decision problem*. Universitext. Springer-Verlag, Berlin, 2001. Reprint of the 1997 original.

[16] T. Braüner. Natural deduction for hybrid logic. *Journal of Logic and Computation*, 14:329–353, 2004. Revised and extended version of paper in *Workshop Proceedings of Methods for Modalities 2*.

[17] T. Braüner. Natural deduction for first-order hybrid logic. *Journal of Logic, Language and Information*, 14:173–198, 2005. Revised and extended version of paper in *Workshop Proceedings of Fourth Workshop on Hybrid Logics*.

[18] T. Braüner. Axioms for classical, intuitionistic, and paraconsistent hybrid logic. *Journal of Logic, Language and Information*, 2006. To appear.

[19] G. Chierchia and S. McConnell-Ginet. *Meaning and Grammar. An Introduction to Semantics*. MIT Press, Cambridge, Mass., 1990.

[20] N.B. Cocchiarella. Philosophical perspectives on quantification in tense and modal logic. In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, 2nd Edition*, volume 7, pages 235–275. Kluwer Academic Publishers, 2002.

[21] J. Copeland, editor. *Logic and Reality: Essays in the Legacy of Arthur Prior*. Oxford University Press/Clarendon Press, 1996.

[22] G. Corsi. Quantified modal logics of positive rational numbers and some related systems. *Notre Dame J. Formal Logic*, 34(2):263–283, 1993.

[23] G. Corsi. BF, CBF and Lewis semantics. *Logique et Analyse*, 2004. to appear.

[24] G. Corsi and S. Ghilardi. Directed frames. *Archive for Mathematical Logic*, 29:53–67, 1989.

[25] G. Corsi and S. Ghilardi. Semantical aspects of quantified modal logic. In *Knowledge, belief, and strategic interaction (Castiglioncello, 1989)*, Cambridge Stud. Probab. Induc. Decis. Theory, pages 167–195. Cambridge Univ. Press, Cambridge, 1992.

[26] D. de Jongh and G. Dzhaparidze. The Logic of Provability. In S.R. Buss, editor, *Handbook of Proof Theory.* Studies in Logic and the Foundations of Mathematics, Vol.137., pages 475–546. Elsevier, Amsterdam, 1998.

[27] A. G. Drágalin. *Mathematical intuitionism*, volume 67 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, RI, 1988. Introduction to proof theory, Translated from the Russian by Elliott Mendelson, Translation edited by Ben Silver.

[28] M. Fara and T. Williamson. Counterparts and actuality. *Mind*, 114, 2005. to appear.

[29] C. Fermüller, A. Leitsch, T. Tammet, and N. Zamov. *Resolution methods for the decision problem*, volume 679 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1993. Lecture Notes in Artificial Intelligence.

[30] K. Fine. Model theory for modal logic. I. The de re/de dicto distinction. *Journal of Philosophical Logic*, 7:125–156, 1978.

[31] K. Fine. Model theory for modal logic. II. The elimination of de re modality. *Journal of Philosophical Logic*, 7:277–306, 1978.

[32] K. Fine. Failures of the interpolation lemma in quantified modal logic. *Journal of Symbolic Logic*, 44:201–206, 1979.

[33] K. Fine. Model theory for modal logic. III. Existence and predication. *Journal of Philosophical Logic*, 10:293–307, 1981.

[34] K. Fine. The permutation principle in quantificational logic. *Journal of Philosophical Logic*, 12:33–37, 1983.

[35] M. Fitting. A modal logic analog of Smullyan's fundamental theorem. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 19:1–16, 1973.

[36] M. Fitting. Basic modal logic. In D. Gabbay et al., editor, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 1, pages 365–448. Oxford University Press, Oxford, 1993.

[37] M. Fitting. A modal Herbrand theorem. *Fundamenta Informaticae*, 28:101–122, 1996.

[38] M. Fitting. Herbrand's theorem for a modal logic. In A. Cantini, E. Casari, and P. Minari, editors, *Logic and Foundations of Mathematics*, pages 219–225. Kluwer Academic Publishers, 1999.

[39] M. Fitting. *Types, Tableaus, and Gödel's God*. Kluwer, 2002.

[40] M. Fitting. First-order intensional logic. *Annals of Pure and Applied Logic*, 127:171–193, 2004. Essays in the memory of Alfred Tarski. Parts IV, V and VI.

[41] M. Fitting and R.L. Mendelsohn. *First-Order Modal Logic*. Kluwer, 1998.

[42] G. Forbes. *The Metaphysics of Modality*. Clarendon Press, Oxford, 1985.

[43] D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects (Volume 1)*. Oxford University Press, 1994.

[44] D. Gabbay, V. Shehtman, and D. Skvortsov. *Quantification in Nonclassical Logic*. Preliminary draft available at D. Gabbay's homepage http://www.dcs.kcl.ac.uk/staff/dg/.

[45] D. M. Gabbay. Decidability of some intuitionistic predicate theories. *Journal of Symbolic Logic*, 37:579–587, 1972.

[46] D. M. Gabbay and V.B. Shehtman. Undecidability of modal and intermediate first-order logics with two individual variables. *Journal of Symbolic Logic*, 58:800–823, 1993.

[47] D.M. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*, volume 148 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2003.

[48] D. Gallin. *Intensional and Higher-order Modal Logic*. North-Holland, Amsterdam, 1975.

[49] J.W. Garson. Quantification in modal logic. In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, 2nd Edition*, volume 3, pages 267–323. Kluwer Academic Publishers, 2001.

[50] S. Ghilardi. Presheaf semantics and independence results for some non classical first order logics. *Archive for Mathematical Logic*, 29:125–136, 1989.

[51] S. Ghilardi. *Modalità e Categorie*. PhD thesis, Milano, 1990. (English translation in preparation).

[52] S. Ghilardi. Incompleteness results in Kripke semantics. *Journal of Symbolic Logic*, 56(2):517–538, 1991.

[53] S. Ghilardi. Quantified extensions of canonical propositional intermediate logics. *Studia Logica*, 51:195–214, 1992.

[54] S. Ghilardi. Substitution, quantifiers and identity in modal logic. In *New Essays in Free Logic*, volume 23 of *Applied Logic Series*, pages 87–115. Kluwer, 2001.

[55] S. Ghilardi and G. Meloni. Modal and tense predicate logic: models in presheaves and categorical conceptualization. In *Categorical algebra and its applications*, volume 1348 of *Lecture Notes in Mathematics*, pages 130–142. Springer, 1988.

[56] S. Ghilardi and G. Meloni. Relational and topological semantics for modal and temporal first order predicative logic. In *Nuovi problemi della logica e della filosofia della scienza, vol. 2*, pages 59–77. CLUEB, Bologna, 1991.

[57] Silvio Ghilardi, Enrica Nicolini, and Daniele Zucchelli. A comprehensive framework for combined decision procedures. In *Proc. of the 5th Int. Work. on Frontiers of Combining Systems (FROCOS-05)*, volume 3717 of *Lect. Not. in Art. Int.* Springer-Verlag, 2005. (extended version available as Technical Report n.304-05, Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano).

[58] R. Girle. *Modal Logics and Philosophy.* Acumen, 2000.

[59] R. Goldblatt. *Logics of Time and Computation.* CSLI Publications, 1987.

[60] E. Grädel. Decision procedures for guarded logics. In *Proceedings of CADE-16*, volume 1632 of *Lecture Notes in Computer Science*, pages 31–51. Springer, 1999.

[61] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic.* MIT Press, 2000.

[62] P. Hasle, P. Øhrstrøm, T. Braüner, and J. Copeland, editors. *Revised and Expanded Edition of Arthur N. Prior: Papers on Time and Tense.* Oxford University Press, 2003. 342 pages.

[63] A. Hazen. Expressive completeness in modal language. *Journal of Philosophical Logic*, 5:25–46, 1976.

[64] A. Hazen. Counterpart theoretic semantics for modal logic. *The Journal of Philosophy*, (76):319–338, 1979.

[65] H.T. Hodes. Some theorems on the expressive limitations of modal languages. *Journal of Philosophical Logic*, 13:13–26, 1984.

[66] G.E. Hughes and M.J. Cresswell. *An Introduction to Modal Logic.* Methuen, 1968.

[67] G.E. Hughes and M.J. Cresswell. *A Companion to Modal Logic.* Methuen, 1984.

[68] G.E. Hughes and M.J. Cresswell. *A New Introduction to Modal Logic.* Routledge, 1996.

[69] T. Jager. An actualist semantics for quantified modal logic. *Notre Dame Journal of Formal Logic*, 23:335–349, 1982.

[70] P.T. Johnstone. *Sketches of an Elephant (A Topos Theory Compendium)*, volume 43–44 of *Oxford Logic Guides.* Clarendon Press, 2002.

[71] R. Kontchakov, A. Kurucz, and M. Zakharyaschev. Undecidability of first-order intuituionistic and modal logics with two variables. 2004. manuscript.

[72] M. Kracht and O. Kutz. The semantics of modal predicate logic I. Counterpart-frames. In F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev, editors, *Advances in Modal Logic, Volume 3*, pages 299–320. World Scientific, 2002.

[73] S. Kripke. The undecidability of monadic modal quantificational theory. *Z. Math. Logik Grundlag. Math.*, 8:113–116, 1962.

[74] S. Kripke. Identity and necessity. In *Identity and Individuation*, pages 135–164. New York University Press, 1971.

[75] S.A. Kripke. Semantical considerations in modal logic. *Acta Philosophica Fennica*, 16:83–94, 1963.

[76] J. Lambek and P. J. Scott. *Introduction to higher order categorical logic*, volume 7 of *Cambridge Studies in Advanced Mathematics.* Cambridge University Press, Cambridge, 1988. Reprint of the 1986 original.

[77] R. Lavendhomme, Th. Lucas, and G.E. Reyes. Formal systems for topos-theoretic modalities. *Bull. Soc. Math. Belg. Sér. A*, 41(2):333–372, 1989.

[78] F.W. Lawvere. Equality in hyperdoctrines and the comprehension schema as an adjoint functor. In *Applications of Categorical Algebra*, pages 1–14. American Mathematical Society, 1970.

[79] D. Lewis. Counterpart theory and quantified modal logic. *Journal of Philosophy*, 65:113–126, 1968.

[80] D. Lewis. Counterparts of persons and their bodies. *Journal of Philosophy*, 68:203–211, 1971.

[81] L. Löwhenheim. Über Möglichkeiten im Relativkalkül. *Math. Annalen*, 76:228–251, 1915.

[82] S. MacLane. *Categories for the working mathematician.* Springer, 1971.

[83] S. MacLane and I. Moerdijk. *Sheaves in Geometry and Logic: a first Introduction to Topos Theory.* Springer-Verlag, 1992.

[84] M. Makkai and G.E. Reyes. Completeness results for intuitionistic and modal logic in a categorical setting. *Annals of Pure and Applied Logic*, 72(1):25–101, 1995.

[85] Michael Makkai and Gonzalo E. Reyes. *First order categorical logic.* Springer-Verlag, Berlin, 1977. Model-theoretical methods in the theory of topoi and related categories, Lecture Notes in Mathematics, Vol. 611.

[86] M. Marx. Tolerance logic. *Journal of Logic, Language and Information*, 10:353–374, 2001.

[87] R. Montague. On the nature of certain philosophical entities. In *Formal Philosophy, Selected Papers of Richard Montague*, pages 148–187. Yale University Press, 1974.

[88] M. Mortimer. On languages with two variables. *Z. Math. Logik Grundlag. Math.*, 21:135–140, 1975.

[89] S. Niefield. Change of base for relational variable sets. *Theory and Applications of Categories*, 12(7):248–261, 2004.

[90] P. Odifreddi. *Classical Recursion Theory*, volume 125 of *Studies in Logic and the Foundations of Mathematics*. North Holland, 1989.

[91] H.J. Ohlbach, A. Nonnengart, M. de Rijke, and D.M. Gabbay. Encoding two-valued non-classical logics in classical logic. In J.A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume 1, pages 1403–1486. MIT Press, 2001.

[92] P. Øhrstrøm and P. Hasle. A.N. Prior's rediscovery of tense logic. *Erkenntnis*, 39:23–50, 1993.

[93] P. Øhrstrøm and P. Hasle. A.N. Prior's logic. In D. Gabbay and J. Woods, editors, *Logic and the Modalities in the Twentieth Century. The Handbook of the History of Logic.*, volume 6. Elsevier, 2005.

[94] H. Ono. A study of intermediate predicate logics. *Publications of the Research Institute for Mathematical Sciences*, 8:619–649, 1972.

[95] H. Ono. On finite linear intermediate predicate logics. *Studia Logica*, 47(4):391–399 (1989), 1988.

[96] Z. Parks. Investigations into quantified modal logic–I. *Studia Logica*, pages 109–125, 1976.

[97] A. Plantinga. Actualism and possible worlds. In M. Davidson, editor, *Essays in the Metaphysics of Modality*, pages 103–121. Oxford University Press, 2003. Originally published in *Theoria*, 42:139–160, 1976.

[98] A. Prior. *Time and Modality*. Clarendon/Oxford University Press, 1957.

[99] A. Prior. *Past, Present and Future*. Clarendon/Oxford University Press, 1967.

[100] A. Prior. *Papers on Time and Tense*. Clarendon/Oxford University Press, 1968.

[101] W.V.O. Quine. Three grades of modal involvement. In *The Ways of Paradox and Other Essays*, pages 156–174. Random House, 1953.

[102] Helena Rasiowa and Roman Sikorski. *The mathematics of metamathematics*. Monografie Matematyczne, Tom 41. Państwowe Wydawnictwo Naukowe, Warsaw, 1963.

[103] G. E. Reyes. A topos-theoretic approach to reference and modality. *Notre Dame J. Formal Logic*, 32(3):359–391, 1991.

[104] G.E. Reyes and H. Zolfaghari. Topos-theoretic approaches to modality. In *Category Theory*, volume 1488 of *Lecture Notes in Mathematics*, pages 359–378. Springer, 1991.

[105] M. N. Rybakov. Enumerability of modal predicate logics and the breaking of increasing chains. In *Logical investigations, No. 8 (Russian) (Moscow, 2001)*, pages 155–167. "Nauka", Moscow, 2001.

[106] M. N. Rybakov. On the algorithmic expressibility of a modal language with only one monadic predicate letter. In *Logical investigations, No. 9 (Russian)*, pages 179–201. "Nauka", Moscow, 2002.

[107] D. Scott. A decision method for for validity of sentences in two variables. *Journal of Symbolic Logic*, 27:477, 1962.

[108] D.S. Scott. Advice on modal logic. In K. Lambert, editor, *Philosophical Problems in Logic*, pages 143–173. D. Reidel Publishing Company, 1970.

[109] K. Segerberg. Two-dimensional modal logic. *Journal of Philosophical Logic*, 2:77–96, 1973.

[110] V.B. Shehtman. On some two-dimensional modal logics. In *8th Congress on Logic Methodology and Philosophy of Science, vol. 1*, pages 326–330. Nauka, Moskow, 1987.

[111] V.B. Shehtman and D.P. Skvortsov. Semantics of non-classical first-order predicate logics. In *Mathematical logic*, pages 105–116. Plenum Press, New York, 1990.

[112] T. Shimura. Kripke completeness of some intermediate predicate logics with the axiom of constant domain and a variant of canonical formulas. *Studia Logica*, 52(1):23–40, 1993.

[113] T. Shimura. Kripke incompleteness of predicate extensions of the modal logics axiomatized by a canonical formula for a frame with a non-trivial cluster. *Studia Logica*, 65:237–247, 2000.

[114] H. Shirasu. Duality in superintuitionistic and modal predicate logics. In *Advances in Modal Logic I*, number 87 in CSLI Lecture Notes, pages 223–236. CSLI Publ., Stanford, CA, 1998.

[115] D. P. Skvortsov. On the predicate logics of finite Kripke frames. *Studia Logica*, 54(1):79–88, 1995.

[116] D. P. Skvortsov. Not every "tabular" predicate logic is finitely axiomatizable. *Studia Logica*, 59(3):387–396, 1997.

[117] D. P. Skvortsov. On intermediate predicate logics of some finite Kripke frames. I. Levelwise uniform trees. *Studia Logica*, 77(3):295–323, 2004.

[118] D. P. Skvortsov. On the predicate logics of linear Kripke frames and some its extensions. *Studia Logica*, 2005. to appear.

[119] D.P. Skvortsov. On axiomatizability of some intermediate predicate logics. *Reports on Mathematical Logic*, 22:115–116, 1988.

[120] D.P. Skvortsov. The superintuitionistic predicate logic of finite Kripke frames is not recursively axiomatizable. *Journal of Symbolic Logic*, 2005. to appear.

[121] D.P. Skvortsov and V.B. Shehtman. Maximal Kripke-type semantics for modal and superintuitionistic predicate logics. *Annals of Pure and Applied Logic*, 63:69–101, 1993.

[122] R. Stalnaker and R. Thomason. Abstraction in first-order modal logic. *Theoria*, 34:203–207, 1968.

[123] H. Sturm and F. Wolter. First-order expressivity for S5-models: Modal vs. two-sorted languages. *Journal of Philosophical Logic*, 30:571–591, 2001.
[124] J. Surańy. Zur Reduktion des Entscheidungsproblem des logischen Funktionskalcüls. *Matematika Fizika Lepok*, 50:51–74, 1943.
[125] Y. Tanaka and H. Ono. Rasiowa-Sikorski lemma and Kripke completeness of predicate and infinitary modal logics. In *Advances in Modal Logic II*, number 119 in CSLI Lecture Notes, pages 401–419. CSLI Publ., Stanford, CA, 2001.
[126] R. Thomason and R. Stalnaker. Modality and reference. *Nous*, 2:359–372, 1968.
[127] J. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, 1983.
[128] J. van Benthem. Beyond accessibility: Functional models for modal logic. In M. de Rijke, editor, *Diamonds and Defaults*, pages 1–18. Kluwer Publ., Dordrecht, The Netherlands, 1993.
[129] J. van Benthem. Correspondence theory. In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, 2nd Edition*, volume 3, pages 325–408. Kluwer Academic Publishers, 2001.
[130] F. Wolter and M. Zakharyaschev. Decidable fragments of first-order modal logics. *Journal of Symbolic Logic*, 66:1415–1438, 2001.
[131] S. Yokota. Axiomatization of the first-order intermediate logics of bounded kripkian heights, 1. *Z. Math. Logik Grundlag. Math.*, 35(5):414–421, 1989.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

10

HIGHER ORDER MODAL LOGIC

Reinhard Muskens

1 Introduction . . . 621
2 Motivation . . . 622
2.1 The Ontological Argument . . . 622
2.2 Axiomatization of Classical Mechanics . . . 623
2.3 The Semantics of Natural Language . . . 623
2.4 Modal Logics with Propositional Quantifiers . . . 626
3 Montague's Intensional Logic . . . 627
3.1 Overview of **IL** . . . 627
3.2 Limitations of **IL** . . . 629
4 A Modal Type Theory . . . 631
4.1 Types and Terms . . . 632
4.2 Standard Models . . . 634
4.3 Intensional Models . . . 635
5 Tableaus for Modal Type Theory . . . 638
5.1 Tableaus . . . 638
5.2 Axiomatic Extensions . . . 640
6 Elementary Model Theory . . . 642
6.1 Identity and Indiscernability . . . 642
6.2 Model Existence . . . 644
7 Conclusion . . . 649

1 INTRODUCTION

A logic is called *higher order* if it allows for quantification (and possibly abstraction) over higher order objects, such as functions of individuals, relations between individuals, functions of functions, relations between functions, etc. Higher order logic (often also called *type theory* or the *Theory of Types*) began with Frege, was formalized in Russell [46] and Whitehead and Russell [52] early in the previous century, and received its canonical formulation in Church [14].[1] While classical type theory has since long been overshadowed by set theory as a foundation of mathematics, recent decades have shown remarkable comebacks in the fields of *mechanized reasoning* (see, e.g., Benzmüller et

[1]For a good survey of (non-modal) higher order logic, see van Benthem and Doets [8]; for a textbook development, Andrews [3].

al. [9] and references therein) and *linguistics*. Since the late 1960's philosophers and logicians, for various reasons which we will dwell upon, have started to combine higher order logic with modal operators (Montague [35, 37, 38], Bressan [11], Gallin [22], Fitting [19]). This combination results in *higher order modal logic*, the subject of this chapter.

The chapter will be set up as follows. In the next section we will look at possible motivations behind the idea of combining modality and higher order logic. Then, in Section 3, Richard Montague's system of 'Intensional Logic', by far the most influential of higher order modal logics to date, will be discussed. This logic will be shown to have some limitations. One of these is that, despite its name, the logic is not fully intensional, as it validates the axiom of Extensionality. This leads to a series of well-known problems centering around 'logical omniscience'. Another limitation is that the logic is not Church-Rosser (it matters in which order λ-conversions are carried out). These limitations can be overcome and the remaining sections of the chapter will contain an exposition of a modal type theory that is intensional in two ways: in the sense of being a modal logic and in the sense that Extensionality does not hold. The logic in itself is not strong enough to make the usual rules of λ-conversion derivable, but these rules can consistently be added as an axiomatic extension and in that case the Church-Rosser property will hold (as an alternative, the rules can be hard-wired into the theory, in which case the theory is also Church-Rosser). Section 4 will introduce the basic syntax and semantics of this logic, Section 5 will give a tableau calculus, and Section 6 provides some elementary model theory in the form of a model existence theorem and its usual corollaries, such as generalized completeness. We conclude with a conclusion.

2 MOTIVATION

Why should one want to combine modality with quantification or abstraction over objects of higher type? Possible reasons come from areas as diverse as rational theology, the axiomatization of classical mechanics, the semantics of natural language, and modal logic itself. Let us look at each of these in turn.

2.1 The Ontological Argument

Anselm (1033–1109) proved the existence of God by defining him as "a being than which none greater can be thought" and by arguing that, since that definition can be understood, such a being must "exist in the understanding". But if this being exists in the understanding, one can also think of it as existing in reality and, since real existence is "greater" than mere conceptual existence, the "being than which none greater can be thought" must truly exist. Otherwise one could think of an even greater being that did truly exist. Moreover, by an analogous argument, Anselm comes to the conclusion that it is even *impossible* to think of God as nonexistent. For something that cannot be thought of as nonexisting is greater than something that can be so thought of. It follows that a "being than which none greater can be thought" cannot merely exist contingently, otherwise one could think of an even greater being with necessary existence.

Anselm's original argument was phrased in ordinary Latin and its lack of precision may be deemed a weakness by some, but increasingly more precise variants of the argument have been put forward by Descartes, Leibniz and, more recently, Gödel [24]. Gödel's argument centers around "positive" properties and being a god can be defined as having

every positive property. There are axioms regulating the behaviour of the predicate "positive", stipulating e.g. that exactly one of a property or its complement is positive, that, whenever P is positive and the extension of P necessarily is a subset of that of Q, Q is also positive, that necessary existence is a positive property, etc., etc. The conclusion is identical to that of Anselm's: God necessarily exists (and is unique), but this time premises and argument are spelled out in great detail (some of the premises may be hard to swallow, even for those who are willing to accept the conclusion). The argument combines quantification over properties with the modal notions of necessity and possibility and as a consequence is naturally framed in a higher order modal logic. For a recent evaluation of Gödel's proof, its history, a precise formalization, and extensive discussion of the argument and subsequent literature, see Fitting [19].

2.2 Axiomatization of Classical Mechanics

As a second example of the use of a higher order modal logic, let us briefly mention the proposal in Bressan [11] for a logical foundation of classical mechanics in general and Mach's often criticized definition of *mass* in particular. Mach's definition has a counterfactual character and this is where modality comes in. Suppose we have a particle M, whose mass is to be established. Fix some inertial frame. If a particle M_1 with unit mass and velocity v_1 parallel to M's velocity v were to collide with M at time t, then, if the changes in the velocities of M and M_1 at t would be $\Delta v \neq 0$ and Δv_1 respectively, the mass of M would be $\Delta v_1/\Delta v$. This means that the mass of M *can* be established experimentally, but, as Bressan points out, in an axiomatic foundation of physics it is important that the axioms do not imply that the experiment actually takes place, as many physically possible situations that one wants to be able to describe are in fact incompatible with such an assumption. Thus, Bressan argues, an axiomatization based on a modal logic is required. Bressan's logic is not only modal but also higher order, as it essentially replaces set theory and concepts such as natural number and real number should therefore be definable within the logic (for example, the natural number n is defined as the property of having n elements, $\lambda F.\exists_n x\, Fx$, with $\exists_n$ the obvious abbreviation, in Frege's way).

2.3 The Semantics of Natural Language

A third illustration of the use of higher order modal logic comes from Richard Montague's [36, 37, 38] contributions to the semantics of natural language, work that truly revolutionized the subject.[2] It was Montague's aim to treat the semantics of natural language in a completely precise way and to provide a truth definition for sentences of (say) English very much in accordance with the usual Tarski truth definition for logical languages. One way to achieve this is to directly assign modeltheoretic objects to syntactic expressions. This road was taken in Montague [36], but a way that is easier to go in practice is to translate expressions of English to an interpreted logical language. The interpretation of the logic then indirectly provides a model theory for the fragment of English under consideration. This is done in Montague [37, 38]. The logic used is higher order and modal.

[2]The presentation here is inspired by Montague's work but deviates from it in many minor details.

Why did Montague use a logic that combined modality with higher order quantification and abstraction? It is not difficult to see why one should want a *modal* logic for the treatment of natural language, as the latter abounds with phrases and constructions that have motivated modal logics in the first place (temporal operators, counterfactuals, true modals like *can, might* and *would*, propositional attitude verbs, and so on). But the reason for employing a *higher order* logic may be less clear to logicians not working in linguistics. Although natural language is able to quantify over properties and in general can express things that are not normally expressible with first order means only (think of sentences like *most men have green eyes*, for example), this is not the sole or even the primary reason for using type theory in linguistics. The main reason is that in type theory the availability of lambda-abstraction allows for a *closure of the gap between the syntactic forms of natural language expressions and those of their logical translations.*

Let us illustrate this with the help of the simple example sentence in (1a). Linguists almost universally provide this sentence with a constituent structure along the lines of (1b), i.e. the determiner *every* is thought to form a constituent (a noun phrase) with the noun *elephant* and this resulting constituent then forms another constituent (a sentence) with the verb phrase *danced*. Essentially, therefore, the linguistic analysis of such sentences follows the pre-Fregean pattern of dividing each sentence in a subject (here *every elephant*) and a predicate (*danced*).

(1) a. every elephant danced
b. [[every elephant] danced]
c. $\forall x\,(Ex \to Dx)$
d. $(((\lambda P_1 \lambda P_2.\forall x\,(P_1 x \to P_2 x))E)D)$

The analysis of natural language expressions as consisting of larger and larger clusters of constituents is an important feature of modern linguistic theory, and syntacticians are in the possession of a whole battery of empirical tests to determine constituenthood, but the syntactic form that is given to any sentence is not in general congruent with its usual logical form. The structure in (1b), for example, is fundamentally different from that of the logical sentence (1c), the usual translation of (1a). While the constituents *elephant* and *danced* in (1b) reappear in (1c) as E and D respectively, there are no continuous parts of (1c) corresponding to *every* or *every elephant.* This gap between logical form and linguistic form is what logicians such as Russell and Quine had in mind when they alluded to the *misleading form* of natural language: the 'correct' form of (1a) according to this perspective is (1c); (1b) merely misleads. This point of view could never be shared by the linguistic community, as giving up the standard notion of constituenthood would greatly diminish the predictive power of syntactic theory.

Can the gap be bridged? Here lambdas come to the rescue, for in a higher order logic with lambda abstraction (1c) can alternatively be written as (1d). While (1c) is the β-normal form of (1d), the latter, but not the former, follows the syntactic pattern in (1b). Lambdas allow us to have our cake and eat it. They allow us to maintain the view that the logical form of an expression closely mirrors its syntactic form without having to give up the usual logical analysis.

In fact, with lambdas in hand, it is now possible to think of inductive translation mechanisms sending syntactic forms to logical forms. In the present case one can translate *every* as $\lambda P_1 \lambda P_2.\forall x\,(P_1 x \to P_2 x)$, a term containing two λ-abstractions over predicates, *elephant* can be translated as the predicate constant E and *dances* as D. If

one lets onstituent formation correspond to application, [every elephant] translates as $(\lambda P_1 \lambda P_2.\forall x\,(P_1 x \rightarrow P_2 x))E$, which reduces to $\lambda P_2.\forall x\,(Ex \rightarrow P_2 x)$ (a *generalized quantifier*), and a further step shows that (1b) translates as (1d), or, equivalently, (1c).

But now a difficulty crops up. If [every elephant] translates as $\lambda P_2.\forall x\,(Ex \rightarrow P_2 x)$, how are we going to translate the verb phrase [fed [every elephant]] in (2b), the syntactic analysis of (2a)? The verb *fed* should presumably be translated as some binary relation F between individuals and this is not the kind of object that $\lambda P_2.\forall x\,(Ex \rightarrow P_2 x)$ can apply to (or that can apply to that term).

Montague solved this by complicating the translations of transitive verbs like *fed*. He translated *fed* not simply as F, but as the term $\lambda Q \lambda x.Q(Fx)$[3] (with Q ranging over quantifiers and x over individuals), and if the translations for *a* and *girl* are chosen to be $\lambda P_1 \lambda P_2.\exists x\,(P_1 x \wedge P_2 x)$ and G respectively, the translation in (2c) results, as the reader may care to verify.

(2) a. a girl fed every elephant

b. [[a girl][fed [every elephant]]]

c. $\exists x\,(Gx \wedge \forall y\,(Ey \rightarrow Fxy))$

d. $\forall y\,(Ey \rightarrow \exists x\,(Gx \wedge Fxy))$

Translating an intransitive verb like *fed* as $\lambda Q \lambda x.Q(Fx)$, and not as the simpler and more intuitive binary relation symbol F, seems ad hoc, however. In fact, researchers in the Montague tradition have argued that a combination of giving simple translations with providing systematic ways of obtaining certain translations from others is not only more elegant than Montague's original approach was, but also gives a better fit with the data (Partee and Rooth [44], Hendriks [26, 27]). Discussing the calculi for 'shifting' translations that these authors have proposed would lead us too far afield here. Suffice it to say that from their considerations, in conjunction especially with those of van Benthem [7], the picture emerges that *linear combinators*[4] play an all-important role. The translation of *fed* as $\lambda Q \lambda x.Q(Fx)$, for example, can be thought to result from applying the linear combinator $\lambda R \lambda Q \lambda x.Q(Rx)$ to a basic translation F, while applying the combinator $\lambda R \lambda Q_1 \lambda Q_2.Q_1(\lambda y.Q_2(\lambda x.Rxy))$ to F results in a translation that eventually leads to (2d), another possible translation of the original sentence.[5]

For more information on Montague's approach to the semantics of natural language, see the textbooks Dowty et al. [15] and Gamut [23], the survey in Partee with Hendriks [43], and the chapter on Linguistics by Moss and Tiede in this handbook (Chapter 19). Montague's higher order modal logic **IL** will be described shortly.

---

[3] For the sake of exposition I am disregarding Montague's intensional operators here.

[4] A *combinator* is a closed λ-term built from variables with the help of λ-abstraction and application only. A combinator M is *linear* if each abstractor λX in M binds exactly one X in M.

[5] While linear combinators play an important role in semantic composition, just letting them apply to semantic translations without further ado results in serious overgeneration. Applying the permutation operator $\lambda R \lambda y \lambda x.Rxy$ to F above, for example, would allow the derivation of translations for *a girl fed every elephant* that are normally associated with *an elephant fed every girl*. Partee and Rooth [44] and Hendriks [26, 27] provide calculi in which permutation is not derivable, while de Groote [25] and Muskens [40, 41] base their grammars entirely on linear lambda terms but make sure that any permutation in semantics is mirrored by a permutation in syntax.

2.4 Modal Logics with Propositional Quantifiers

Motives inherent in modal logic itself may also lead to a combination of modality with higher order, or at least second order, quantification. The standard definition of the truth of a formula in a frame at a world is defined with the help of a quantification over valuations and therefore essentially corresponds to universal quantification over sets of possible worlds. More precisely, the frame truth of a formula φ containing proposition letters $p_1, \ldots, p_n$ corresponds to the truth of a formula $\forall P_1 \ldots P_n \varphi'$, where φ' is the standard translation $ST(\varphi)$ of φ; see Chapter 1 by Blackburn and van Benthem in this handbook. This gives *global* second order quantification, with the second order universal quantifiers taking scope over the whole formula, but one may now be inspired to add quantifiers $\forall p$ and $\exists p$ ranging over sets of possible worlds to given modal logics. This was done in Kripke [33] and modal logics with propositional quantifiers have been studied by a variety of authors since, amongst whom are Bull [12], Fine [16], Kaplan [30], Kremer [32, 31], Fitting [18], and ten Cate [13], to name but a few.

Semantically there are two lines of attack here. If one has a frame $\langle W, R\rangle$, the most obvious interpretation of quantifiers $\forall p$ and $\exists p$ in that frame lets them range over the power set $\mathcal{P}(W)$ of the set of possible worlds W. This is called the *second order* (or *primary) interpretation* of propositional quantifiers. If propositional quantifiers are added to a modal logic **L** in this way (where **L** = **S4**, **S5**, etc.), the resulting logic is called **L**π+. The behaviour of the logics thus obtained rather varies. **S5**π+, on the one hand, is decidable (Fine [16], Kaplan [30]), as this logic is embeddable into monadic second order logic. (The embedding essentially is the standard translation, with clauses such as $ST(\Box\varphi) = \forall x ST(\varphi)$ and $ST(\forall p\varphi) = \forall P\, ST(\varphi)$.) Fine and Kaplan also axiomatize **S5**π+. The logics **K**π+, **T**π+, **K4**π+, **B4**π+, **S4.2**π+, and **S4**π+, on the other hand, are recursively isomorphic to full second order logic (this was proved independently by Kripke and Fine; Fine [16] has a weaker result).

In order to obtain nice proof systems for modal logics with propositional quantification, one can also follow the example of Henkin [28], who, in the context of higher order logic, defined a class of models in which higher order quantifiers do not necessarily range over *all* subsets of the relevant domains, but only over designated subsets of them. In the present context such a set-up means that frames $\langle W, R\rangle$ are replaced by triples $\langle W, R, \Pi\rangle$ such that $\Pi \subseteq \mathcal{P}(W)$. Here Π must be closed under boolean operations, including arbitrary unions and intersections and it must be the case that $R[P] \in \Pi$ and $R^{-1}[P] \in \Pi$ whenever $P \in \Pi$ (see e.g. Thomason [50], who considers such structures for tense logics). Propositional quantification is now interpreted as quantification over Π. This is the so-called *first order* (or *secondary*) *interpretation* of propositional quantifiers. The resulting logics are denoted as **S4**π, **S5**π, etc., according to the constraints that are put on accessibility relations R. All these logics are axiomatizable with the help of reasonable axioms.

How does the axiomatization of **S5**π that one gets in this way (basically the usual **S5** axioms and rules + the usual quantification axioms and rules for propositional quantification) compare to the one obtained by Fine and Kaplan? Curiously, an axiomatization of **S5**π+ requires one additional axiom, namely

(3) $\exists p(p \wedge \forall q(q \rightarrow \Box(p \rightarrow q)))$

A little reflection shows that if this formula is evaluated in a world w in a frame $\langle W, R\rangle$ using the primary interpretation, it is true, with $\{w\}$ as a sole witness for p. On the other hand, evaluation with respect to w in a frame $\langle W, R, \Pi\rangle$ may not result in truth, as there

may be no $P \in \Pi$ such that $w \in P$ and $P \subseteq P'$ for all P' such that $w \in P' \in \Pi$. A very similar situation obtains in higher order logic. In the models of Henkin [28] sets may be so sparse that there are not enough of them to distinguish between objects that are in fact not identical. Two distinct objects d_1 and d_2 may have exactly the same properties, and in particular $\{d_1\}$ may fail to exist (Andrews [2]). In a modal context definability of singleton sets $\{w\}$ can be enforced through the introduction of *nominals* (Blackburn et al. [10], Areces and ten Cate, Chapter 14 of this handbook).

3 MONTAGUE'S INTENSIONAL LOGIC

In the previous section we explained some of Montague's ideas with the help of a non-modal logic, but Montague himself actually framed them in **IL** (Intensional Logic), a higher order modal logic that will be discussed in this section. (See also Moss and Tiede's Chapter 19 of this handbook. For a highly interesting alternative to Montague's **IL**, see Zalta [53, 54, 55]). The logic is an extension of Church's [14] theory of types and inherits many, though not all, of the latter's properties.

3.1 Overview of **IL**

In order to set up the logic, one first needs to define a simple type system.

DEFINITION 1. The set of **IL** *types* is the smallest set of strings such that:

(i) e and t are **IL** types;

(ii) If α and β are **IL** types, then $(\alpha\beta)$ is an **IL** type;

(iii) If α is an **IL** type, then $(s\alpha)$ is an **IL** type.

Here the type e is the type of *entities*, while t is the type of *truth-values*. Note that, while s can be used to form complex **IL** types, it is not itself an **IL** type. The intended interpretation of the types defined here is that objects of a type $\alpha\beta$ (also written $\alpha \to \beta$) are functions from objects of type α to objects of type β and that objects of type $s\alpha$ are functions from the set of possible worlds to objects of type α.

The next step is to define the *terms* of **IL**. It will be assumed that each **IL** type α comes with a denumerably infinite set of variables and a countable set of constants. Terms are built up from these as follows.

DEFINITION 2. Define, for each **IL** type α, the set of **IL** terms T_α as follows.

(i) Every constant or variable of any type α is an element of T_α;

(ii) If $A \in T_{\alpha\beta}$ and $B \in T_\alpha$, then $(AB) \in T_\beta$;

(iii) If $A \in T_\beta$ and x is a variable of type α then $(\lambda x.A) \in T_{\alpha\beta}$;

(iv) If $A, B \in T_\alpha$ then $(A \equiv B) \in T_t$;

(v) If $A \in T_\alpha$ then $({}^\wedge A) \in T_{s\alpha}$;

(vi) If $A \in T_{s\alpha}$ then $({}^\vee A) \in T_\alpha$.

So we have application and abstraction, identity, and "cap" and "cup" operators that, as we will see, are very much analogous to application and abstraction. If $A \in T_\alpha$ we will often indicate that fact by writing A_α. Terms of type t are called *formulas* and we often use metavariables φ, ψ, etc. to range over them.

Definition 2 does not seem to provide us with the expressivity that we want, as the common logical operators, including the modal $\Box$ and $\Diamond$ seem to be absent, but in fact such operators are definable from the ones just adopted (Henkin [29], Gallin [22]).

DEFINITION 3. Write

$$\begin{aligned}
\top &\text{ for } (\lambda x_t.x) \equiv (\lambda x_t.x),\\
\bot &\text{ for } (\lambda x_t.x) \equiv (\lambda x_t.\top),\\
\neg\varphi &\text{ for } \bot \equiv \varphi,\\
\varphi \wedge \psi &\text{ for } (\lambda f_{tt}.f\varphi \equiv \psi) \equiv (\lambda f_{tt}.f\top),\\
\forall x_\alpha \varphi &\text{ for } (\lambda x_\alpha.\varphi) \equiv (\lambda x_\alpha.\top), \text{ and}\\
\Box\varphi &\text{ for } {}^\wedge\varphi \equiv {}^\wedge\top.
\end{aligned}$$

Other operators will have their usual definitions.

Whether these abbreviations make sense can be checked as soon as we are in the possession of a semantics for the language. So let us turn to that.

DEFINITION 4. A *(standard) model* for **IL** is a triple $\langle D, W, I\rangle$ such that D and W are non-empty sets and I is a function with the set of all constants as its domain, such that $I(c) \in D_{s\alpha}$ for each constant c of type α, where the sets D_α are defined using the following induction.

$$\begin{aligned}
D_e &= D\\
D_t &= \{0, 1\}\\
D_{\alpha\beta} &= \{F \mid F : D_\alpha \to D_\beta\}\\
D_{s\alpha} &= \{F \mid F : W \to D_\alpha\}.
\end{aligned}$$

The function I is called an *interpretation* function. Intuitively, we interpret D as a domain of possible individuals and W as a set of possible worlds.

In order to interpret terms on models, we additionally need to define an *assignment* to $M = \langle D, W, I\rangle$ as a function a with the set of all variables as its domain, such that $a(x) \in D_\alpha$ if x is of type α. The notation $a[d/x]$ is defined as usual. Terms can now be evaluated on models with the help of a Tarski-style truth definition.

DEFINITION 5. The *value* $\|A\|^{M,w,a}$ of a term A on a model $M = \langle D, W, I\rangle$ in world $w \in W$ under an assignment a to M is defined in the following way:

(i) $\|c\|^{M,w,a} = I(c)(w)$ if c is a constant; $\|x\|^{M,w,a} = a(x)$ if x is a variable;

(ii) $\|AB\|^{M,w,a} = \|A\|^{M,w,a}(\|B\|^{M,w,a})$;

(iii) $\|\lambda x_\beta A\|^{M,w,a}$ = the function F with domain D_β such that $F(d) = \|A\|^{M,w,a[d/x]}$ for all $d \in D_\beta$;

(iv) $\|A \equiv B\|^{M,w,a} = 1$ iff $\|A\|^{M,w,a} = \|B\|^{M,w,a}$;

(v) $\|{}^\wedge A\|^{M,w,a}$ = the function F with domain W such that $F(w') = \|A\|^{M,w',a}$ for all $w' \in W$;

(vi) $\|{}^\vee A\|^{M,w,a} = \|A\|^{M,w,a}(w)$;

Note the special treatment of the non-logical constants in the first clause of this definition: constants of type α are interpreted as functions of type $s\alpha$ by the interpretation function I but these functions are applied to the current world in order to get the actual value, an object which is of type α again. The second and third clauses interpret application and

abstraction in a way that is to be expected. The fourth clause interprets $\equiv$ as identity *relative to a possible world*, i.e. $A \equiv B$ means that A and B have the same extension in the world of evaluation, not necessarily in all possible worlds. The last two clauses interpret the cap and cup operators in a way that is analogous to abstraction and application; cap is abstraction over possible worlds while cup is application to the current world. We leave it to the reader to verify that the abbreviations of definition 3 provide the operators defined there with their usual semantics (with $\Box$ the universal modality).

A formula φ is *true* in a model M in world w under an assignment a if $\|\varphi\|^{M,w,a} = 1$. The notion of standard entailment, or s-entailment for short, is defined accordingly.

DEFINITION 6. Let Γ and Δ be sets of **IL** formulae. We say that Γ *s-entails* Δ, $\Gamma \models_s \Delta$, if, whenever $M = \langle D, W, I\rangle$ is a model, $w \in W$, and a is an assignment to M, $\|\varphi\|^{M,w,a} = 1$ for all $\varphi \in \Gamma$ implies $\|\psi\|^{M,w,a} = 1$ for some $\psi \in \Delta$.

While it is clear from Gödel's incompleteness theorem that the relation $\models_s$ can have no recursive axiomatization, it is possible to define a *generalized* notion of entailment $\models_g$ that can be so axiomatized. For Church's logic this was done in Henkin [28], while Gallin [22] (in general a rich source of information about Montague's logic) generalizes the completeness proof found there to the setting of **IL**. The $\models_g$ notion is obtained with the help of *generalized* (or: *Henkin*) *models*, the main difference between these and standard models being that, while for each α and β it must hold that $D_{\alpha\beta} \subseteq \{F \mid F : D_\alpha \to D_\beta\}$, the $D_{\alpha\beta}$ need not be the *entire* function spaces $\{F \mid F : D_\alpha \to D_\beta\}$. Similarly, it is only required that $D_{s\alpha} \subseteq \{F \mid F : W \to D_\beta\}$. We will not pursue the proof of Henkin (or: generalized) completeness for **IL** here, but refer to Gallin's original work. For a generalized completeness proof for a similar higher order modal logic, see Section 6.

3.2 Limitations of **IL**

Montague's work has been a tremendous boost for natural language semantics but with the advantage of hindsight it is possible to point out some shortcomings of the logic that he used. These limitations will be reviewed here. First, let us ask ourselves the question whether the logic lives up to its name. Is **IL** really an *intensional* logic? If "intensional" merely is another word for "modal" there can be no discussion, but there is an older definition of the concept of intensionality that makes perfect sense in a higher order context and in which sense **IL** is not intensional. Whitehead and Russell's *Principia Mathematica* [52, number *20] is one place where this definition can be found. In this work a distinction between *extensional* and *intensional* functions of functions is made and Whitehead and Russell give as "the mark of an extensional function f" a condition which in their notation reads

(4) $\varphi!x. \equiv_x .\psi!x :\supset_{\varphi,\psi}: f(\varphi!\hat{z}). \equiv .f(\psi!\hat{z})$

but which in the present setting can be written as

(5) $\forall gh(\forall x(gx \equiv hx) \to fg \equiv fh)$

Thus a function of functions f is extensional if, whenever f is applied to a function g, the resulting value fg depends only on the extension of g; a function of functions is intensional if not extensional.[6]

[6]Whitehead and Russell only consider *propositional* functions and as a consequence their f, if it had been typed in our way, would have received a type of the form $(\alpha t)t$ (so that $\equiv$ can be read as $\leftrightarrow$). In

Whitehead and Russell point out that contexts of propositional attitude such as "I believe that p" are examples of functions that are not extensional and hence intensional. However, it is immediately clear that in **IL** all functions of functions are extensional in the sense of (5) and that intensional functions are ruled out. **IL** conforms to the following form of the axiom of Extensionality:

(6) $\forall f \forall gh(\forall x(gx \equiv hx) \to fg \equiv fh)$

For an Intensional Logic this seems below par. The situation is alleviated in a sense by the fact that the following scheme (in which $\varphi\{P := F\}$ denotes the result of substituting the constant F for the variable P in φ) is not generally valid.

(7) $\forall x(Fx \equiv Hx) \to (\varphi\{P := F\} \equiv \varphi\{P := H\})$

For example, one does not have

(8) $\forall x(Fx \equiv Hx) \to (\Box(H \equiv F) \equiv \Box(H \equiv H))$,

as it is easy to construct a model in which H and F are coextensive at some point but not at another. This is desirable, since from the premise that all and only humans are featherless bipeds (to take a truly Russellian example) it should not follow that being a featherless biped necessarily is being human.

But now there is room for a second point of criticism, for how come (6) can be valid while (7) is not? Surely, one can always instantiate g as the constant F, h as H, f as $\lambda P.\, \varphi$ and from

(9) $\forall x(Fx \equiv Hx) \to ((\lambda P.\varphi)F \equiv (\lambda P.\varphi)H)$

get (7) with the help of two β-conversions? The answer is that β-conversion unfortunately is not generally valid in **IL** but is subject to side conditions additional to the usual constraint on substitutability. $(\lambda P.\Box(H \equiv P))F$, for example, is not semantically equivalent to $\Box(H \equiv F)$, as the reader may care to verify.

We will turn to the side conditions on β-conversion shortly, but first, as a third criticism, let us notice that, while the scheme in (7) is not valid, the strengthened version in (10) does hold in all models at any possible world (the proof is by induction on the complexity of φ).

(10) $\Box\forall x(Fx \equiv Hx) \to (\varphi\{P := F\} \equiv \varphi\{P := H\})$

But this is far from desirable. Read "is provable with the help of Zorn's Lemma" for F and "is provable with the help of the Axiom of Choice" for H while choosing "John believes that Zorn's Lemma is P" for φ. It is presumably a necessary fact that everything that is provable with the help of Zorn's Lemma is provable with the help of the Axiom of Choice and vice versa. But from "John believes that Zorn's Lemma is provable from Zorn's Lemma" one cannot conclude "John believes that Zorn's Lemma is provable from the Axiom of Choice". Hence (10) should in fact not be valid. This is what is usually called the problem of logical omniscience but is really a consequence of one variant of the Extensionality principle.

Let us consider the side conditions on β-conversion in **IL**. They will unfortunately lead to a fourth problem. Define a term to be *modally closed* if it is built up from variables and terms of the form $\hat{\ }A$ with the help of application, λ-abstraction and $\equiv$. The following scheme is valid.

IL the scheme in (5) will be valid for f of any type $(\alpha\beta)\gamma$ (with g and h of type $\alpha\beta$ and x of type α).

(11) $(\lambda x_\alpha A_\beta)B_\alpha \equiv A\{x := B\}$, if

(a) B is free for x in A, and

(b) either no free occurrence of x in A lies within the scope of $\hat{}$ or B is modally closed.

This is in fact one of the six axiom schemes that are used to axiomatize generalized consequence in Gallin [22]. But, as was observed by Friedman and Warren [21], the second side condition that needs to be imposed here destroys one of the attractive properties that lambda calculi usually have. For notions of reduction $\twoheadrightarrow$ such as $\twoheadrightarrow_\beta$ or $\twoheadrightarrow_{\beta\eta}$ (see Barendrecht [4] for definitions), one can often establish that whenever $A \twoheadrightarrow A_1$ and $A \twoheadrightarrow A_2$ there is an A_3 such that $A_1 \twoheadrightarrow A_3$ and $A_2 \twoheadrightarrow A_3$, i.e. it is immaterial in which order reductions are made. This so-called *Church-Rosser* property is not retained in **IL** as Friedman and Warren show with the help of (12).

(12) $(\lambda x_\alpha(\lambda y_\alpha.\hat{}y \equiv f_{\alpha(s\alpha)}x)x)c_\alpha$

Here x, y, and f are variables, while c is a constant. One possible reduction leads to

(13) $(\lambda y.\hat{}y \equiv fc)c$,

while another reduction of (12) results in

(14) $(\lambda x.\hat{}x \equiv fx)c$.

Neither of these terms can be reduced any further (as c is not modally closed but the variable that is abstracted over occurs in the scope of $\hat{}$) and hence there is no single term to which both reduce.

Gallin [22] gives a translation of **IL** into a two-sorted variant $\mathbf{TY}_2$ of Church's original logic, which has an extra type s for possible worlds. The translation proceeds by letting $\hat{}$ correspond to λ-abstraction over a fixed variable x_s, while $\check{}$ corresponds to application to x_s (the translation is related to the standard translation of modal logic into first order logic). Constants are translated as the result of application of a constant to the fixed type s variable. This translation clarifies the behaviour of **IL** in many ways. For example, since a term that is not modally closed will translate to a term containing a free occurrence of x_s, the side condition (ii) in (11) in a sense reduces to side condition (i) after all. Since the logic $\mathbf{TY}_2$ is just Church's logic (but two-sorted), it is Church-Rosser, but the difficulty of not being intensional is shared between **IL** and $\mathbf{TY}_2$.

4 A MODAL TYPE THEORY

In the previous section Montague's logic **IL** was described and various criticisms were levelled against it. In this and the next few sections we will propose a logic **MTT** that is compatible with the usual (α), (β) and (η) rules and that is intensional in the sense that two relations can have the same extension yet be different. In order to obtain this logic we must deviate from **IL** in two respects. First, we shall follow Bressan [11] in letting the value of an expression AB in some world w depend not only on (the value of A and) the value of B in w, but possibly on the values of B in other worlds as well. This

immediately solves the problem with β-conversion, as no extra side conditions on that rule will then be necessary.[7]

For the second deviation from **IL**, and indeed from the usual semantics for Church's [14] classical type logic, a class of models will be considered that is a further generalization of the generalized models considered in Henkin [28]. These *intensional models*, as they will be called here, derive from the structures considered in the proofs of cut elimination in Takahashi [49] and Prawitz [45]. The latter also play an important role in Andrews' [1] proof that his (non-extensional) resolution calculus corresponds to the first six axioms of Church [14]. The structures considered by these authors are proof-generated and are defined on the basis of a purely syntactic notion (Schütte's [47] semivaluations), but recently purely semantic, stand-alone, generalizations of such models have been offered in Fitting [19] ('generalized Henkin models') and in Benzmüller et al. [9] ('Σ-models'). Fitting's models involve a non-standard interpretation of abstraction, while the models of Benzmüller et al. have a non-standard form of application, but these complications seem unnecessary, as our intensional models will do without them.

Intensional models will serve two purposes. The first is that they deal with problems of logical omniscience. A second use is technical: the notion of entailment one gets from intensional models is easily axiomatized with the help of a cut free tableau calculus. This second point will be dwelled upon below; for the first point consider the following example. While it is reasonable to assume that sentences (15a) and (15b) determine the same set of possible worlds, it is not reasonable to assume that applying the function "Mary knows that p" to (15a) necessarily results in the same value as applying that function to (15b): (15c) might be true while (15d) is false. Intensional models provide a way to make the necessary distinction. The idea will be that co-entailment, or, more generally, having the same extension in all models, will not imply identity, i.e. the axiom of Extensionality will not hold.

(15) a. The cat is out if the dog is in
b. The dog is out if the cat is in
c. Mary knows that the cat is out if the dog is in
d. Mary knows that the dog is out if the cat is in

4.1 Types and Terms

Unlike **IL**, which is based on hierarchies of *functions*, the logic **MTT** will be based on hierarchies of *relations* (Orey [42], Schütte [47]), as relational models are pleasant to work with. Some definitions therefore must be changed and we shall start with the definition of *types*. Assume that some set $\mathcal{B}$ of *basic types*, among which must be the type s of possible worlds, is given.

DEFINITION 7. The set $\mathcal{T}$ of *types* is the smallest set of strings over the alphabet $\mathcal{B} \cup \{\rangle, \langle\}$ such that (i) $\mathcal{B} \subseteq \mathcal{T}$ and (ii) if $\alpha_1, \ldots, \alpha_n \in \mathcal{T}$ $(n \geq 0)$ then $\langle \alpha_1 \ldots \alpha_n \rangle \in \mathcal{T}$.

Types formed with clause (ii) of this definition will be called *complex*. The complex type $\langle\rangle$, obtained by letting $n = 0$ in (ii), will be the type of *propositions*; this will also be the

[7] See also N. Belnap's foreword to Bressan [11], especially point 11, where this "nonextensional predication" (nonextensional in the modal sense, not in the stronger sense used in this chapter) is called Bressan's cardinal innovation.

type of formulas, which will have sets of possible worlds as their extensions. In general, extensions for terms of type $\langle\alpha_1 \ldots \alpha_n\rangle$ will be $n+1$-ary relations, with one argument place for a possible world (the world where the relation is evaluated) and one for each of the α_i. Note that we have defined types to be certain strings, so that there is a difference between (say) the type s and the type $\langle s\rangle$. The latter is associated with a set of possible worlds in each world, or, equivalently, with the type of binary relations between worlds. Any of these relations can be viewed as an *accessibility* relation.

A *language* will be a countable set of non-logical constants such that each constant has a unique type. If $\mathcal{L}$ is a language, the set of constants from $\mathcal{L}$ having type α is denoted $\mathcal{L}_\alpha$. For each $\alpha \in \mathcal{T}$ we assume the existence of a denumerably infinite set $\mathcal{V}_\alpha$ of variables of type α, such that $\mathcal{V}_\alpha \cap \mathcal{V}_\beta = \varnothing$ if $\alpha \neq \beta$. We let $\mathcal{V} = \bigcup_{\alpha\in\mathcal{T}} \mathcal{V}_\alpha$. In proofs it will occasionally be useful to be able to refer to fixed well-orderings $<_\mathcal{L}$ and $<_\mathcal{V}$ on languages $\mathcal{L}$ and on the set $\mathcal{V}$ respectively, so we will assume that these are in place as well.

The following definition gives terms in all types. Apart from variables and non-logical vocabulary, there will be application and abstraction, and a basis for defining the usual connectives and quantifiers. Moreover, for any term R of type $\langle s\rangle$ there will be a modal operator $\langle R\rangle$ and a term $R^\smile$ intended to denote the converse of R.

DEFINITION 8. Let $\mathcal{L}$ be a language. Define sets $T^\mathcal{L}_\alpha$ of *terms* of $\mathcal{L}$ of type α, for each $\alpha \in \mathcal{T}$, as follows.

(i) $\mathcal{L}_\alpha \subseteq T^\mathcal{L}_\alpha$ and $\mathcal{V}_\alpha \subseteq T^\mathcal{L}_\alpha$ for each $\alpha \in \mathcal{T}$

(ii) If $A \in T^\mathcal{L}_{\langle\beta\alpha_1\ldots\alpha_n\rangle}$ and $B \in T^\mathcal{L}_\beta$, then $(AB) \in T^\mathcal{L}_{\langle\alpha_1\ldots\alpha_n\rangle}$

(iii) If $A \in T^\mathcal{L}_{\langle\alpha_1\ldots\alpha_n\rangle}$ and $x \in \mathcal{V}_\beta$, then $(\lambda x.A) \in T^\mathcal{L}_{\langle\beta\alpha_1\ldots\alpha_n\rangle}$

(iv) $\bot \in T^\mathcal{L}_{\langle\rangle}$

(v) If $\varphi \in T^\mathcal{L}_{\langle\rangle}$ and $\psi \in T^\mathcal{L}_{\langle\rangle}$ then $\varphi \to \psi \in T^\mathcal{L}_{\langle\rangle}$

(vi) If $\varphi \in T^\mathcal{L}_{\langle\rangle}$ and $x \in \mathcal{V}_\alpha$ then $\forall x\varphi \in T^\mathcal{L}_{\langle\rangle}$

(vii) If $R \in T^\mathcal{L}_{\langle s\rangle}$ and $\varphi \in T^\mathcal{L}_{\langle\rangle}$ then $\langle R\rangle\varphi \in T^\mathcal{L}_{\langle\rangle}$

(viii) If $R \in T^\mathcal{L}_{\langle s\rangle}$ then $R^\smile \in T^\mathcal{L}_{\langle s\rangle}$

The operation of taking converses will be useful in applications where the notion arises naturally, such as in temporal logic where, if $<$ is used to denote the relation of temporal precedence, $\langle<\rangle$ will be Prior's future operator F and $\langle<^\smile\rangle$ (or $\langle>\rangle$ after an obvious abbreviation) his past operator P.

We will write $T^\mathcal{L}$ for the set of all terms of the language $\mathcal{L}$, i.e. for the union $\bigcup_{\alpha\in\mathcal{T}} T^\mathcal{L}_\alpha$. If A is a term of type α, we may indicate this by writing A_α and we will use φ, ψ, χ for terms of type $\langle\rangle$, i.e. formulas. The notions *free* and *bound* occurrence of a variable and the notion B *is free for* x *in* A are defined as usual, as are *closed* terms and *sentences*. *Substitutions* are functions σ from variables to terms such that $\sigma(x)$ has the same type as x. If σ is a substitution then the substitution σ' such that $\sigma'(x) = A$ and $\sigma'(y) = \sigma(y)$ for all $y \not\equiv x$ is denoted as $\sigma[x := A]$. If A is a term and σ is a substitution, $A\sigma$, the extension of σ to A, is defined in the usual way. The substitution σ such that $\sigma(x_i) = A_i$ and $\sigma(y) = y$ if $y \notin \{x_1, \ldots, x_n\}$ is written as $\{x_1 := A_1, \ldots, x_n := A_n\}$.

Parentheses in terms will often be dropped on the understanding that association is to the left, i.e. ABC is $((AB)C)$. The operators $\top$, $\neg$, $\wedge$, $\vee$, $\leftrightarrow$ and $\exists$ are obtained as

usual. The following definition gives some other useful operators.

DEFINITION 9. We will write

$$\begin{aligned} =_{\langle\alpha\alpha\rangle} &\text{ for } \lambda x_\alpha \lambda y_\alpha \forall z_{\langle\alpha\rangle}(zx \to zy), \\ [R]\varphi &\text{ for } \neg\langle R\rangle\neg\varphi, \\ \Diamond\varphi &\text{ for } \langle\lambda x_s.\top\rangle\varphi, \\ \Box\varphi &\text{ for } [\lambda x_s.\top]\varphi, \text{ and} \\ \dot{A} &\text{ for } \forall x_{\langle s\rangle}[x]x^\smile A_s. \end{aligned}$$

The first of these abbreviations gives equalities of type $\langle\alpha\alpha\rangle$ for each α. Of course we will usually write $A = B$ instead of $=AB$. The second abbreviation introduces the usual dual to $\langle\cdot\rangle$ and, for example, allows us to write $[<]$ for Prior's G and $[>]$ for his H. The second and third conventions let us write $\Diamond$ and $\Box$ for the *global* possibility and necessity operators, which have the universal relation on worlds as their underlying accessibility relation. The abbreviation $\dot{A}$, lastly, introduces what are called *nominals* (see Blackburn et al. [10] or Areces and ten Cate, Chapter 14 of this handbook, for much more on these). As will become apparent below, $\dot{A}$ will be true in a world w if and only if w is denoted by A.

4.2 Standard Models

Before we introduce the intensional models that will interpret **MTT** terms, let us have a brief look at a class of models that, in order to conform to general usage, will be called *standard* (even though for many practical purposes the intensional models defined below will be preferred).

DEFINITION 10. A *standard collection of domains* is a set $D = \{D_\alpha \mid \alpha \in \mathcal{T}\}$ such that $D_\alpha \neq \varnothing$ if α is basic, $D_\alpha \cap D_\beta = \varnothing$ if $\alpha \neq \beta$ and α and β are basic, while $D_{\langle\alpha_1\ldots\alpha_n\rangle} = \mathcal{P}(D_s \times D_{\alpha_1} \times \cdots \times D_{\alpha_n})$ for each $D_{\langle\alpha_1\ldots\alpha_n\rangle}$. A *standard model* is a pair $\langle D, J\rangle$ such that $D = \{D_\alpha \mid \alpha \in \mathcal{T}\}$ is a standard collection of domains and J is a function with the set of all constants as its domain, such that $J(c) \in D_\alpha$ for each constant c of type α. J is called the *interpretation* function of $\langle D, J\rangle$.

Letting the interpretation function J send constants of type α directly to D_α diverges from the set-up in **IL**, but is in conformity with Church's original logic. It will bring the behaviour of constants in line with that of free variables.

An *assignment* a for a standard collection of domains $D = \{D_\alpha \mid \alpha \in \mathcal{T}\}$ is a function which has the set of variables $\mathcal{V}$ as domain and has the property that $a(x) \in D_\alpha$ if $x \in \mathcal{V}_\alpha$. The usual notational conventions for assignments obtain: If a is an assignment, $x_1, \ldots, x_n$ are pairwise distinct variables, and $d_1, \ldots, d_n$ are objects such that $d_i \in D_\alpha$ if x_i is of type α, then $a[d_1/x_1, \ldots, d_n/x_n]$ is the assignment a' defined by letting $a'(x_i) = d_i$ and $a'(y) = a(y)$, if $y \notin \{x_1, \ldots, x_n\}$.

When working with hierarchies of relations it is often expedient to have a way of interpreting relations as certain functions. The following definition provides one (compare Muskens [39]).

DEFINITION 11. Let R be an n-ary relation ($n > 0$) and let $0 < k \leq n$. Define the *k-th slice function* $F_R^k(d)$ of R by:

$$F_R^k(d) = \{\langle d_1, \ldots, d_{k-1}, d_{k+1}, \ldots, d_n\rangle \mid \langle d_1, \ldots, d_{k-1}, d, d_{k+1}, \ldots, d_n\rangle \in R\}$$

So $F^k_R(d)$ is the $n-1$-ary relation that is obtained from R by fixing its k-th argument place by d. Note that if R is a relation in $\mathcal{P}(D_s \times D_{\alpha_1} \times \cdots \times D_{\alpha_n})$ its first slice function is a function from possible worlds to relations in $\mathcal{P}(D_{\alpha_1} \times \cdots \times D_{\alpha_n})$ and can therefore be identified with what Montague would call a *relation-in-intension*. This motivated the choice of letting $D_{\langle \alpha_1 \dots \alpha_n \rangle}$ equal $\mathcal{P}(D_s \times D_{\alpha_1} \times \cdots \times D_{\alpha_n})$ in definition 10.

The next definition provides terms with values in standard models. Clauses (i) and (iv)–(viii) will probably not surprise the reader, as they are essentially standard; for clauses (ii) and (iii) *second* slice functions provide motivation. For (ii) lets $V(a, AB)$ be equal to the result of applying the second slice function of $V(a, A)$ to $V(a, B)$, while (iii) defines $V(a, \lambda x_\beta.A)$ as the relation whose second slice function is the function F such that, for all $d \in D_\beta$, $F(d) = V(a[d/x], A)$.

DEFINITION 12. The *value* $V_M(a, A)$ of a term A on a standard model $M = \langle D, J \rangle$ under an assignment a to M is defined as follows (we drop subscripts M):

(i) $V(a, c) = J(c)$ if c is a constant; $V(a, x) = a(x)$ if x is a variable;

(ii) $V(a, AB) = \{\langle w, \vec{d} \rangle \mid \langle w, V(a, B), \vec{d} \rangle \in V(a, A)\}$;

(iii) $V(a, \lambda x_\beta.A) = \{\langle w, d, \vec{d} \rangle \mid d \in D_\beta \text{ and } \langle w, \vec{d} \rangle \in V(a[d/x], A)\}$;

(iv) $V(a, \bot) = \varnothing$;

(v) $V(a, \varphi \to \psi) = D_s - (V(a, \varphi) - V(a, \psi))$;

(vi) $V(a, \forall x_\alpha \varphi) = \bigcap_{d \in D_\alpha} V(a[d/x], \varphi)$;

(vii) $V(a, \langle R \rangle \varphi) = \{w \mid \exists w' \in V(a, \varphi) \text{ such that } \langle w, w' \rangle \in V(a, R)\}$;

(viii) $V(a, R^\smile) = \{\langle w, w' \rangle \mid \langle w', w \rangle \in V(a, R)\}$.

4.3 Intensional Models

Intensional models generalize the standard models just given in two ways. The first generalization follows Henkin [28] in not necessarily associating domains $D_{\langle \alpha_1 \dots \alpha_n \rangle}$ with the full powerset $\mathcal{P}(D_s \times D_{\alpha_1} \times \cdots \times D_{\alpha_n})$, but to be contented with some subset of this relational space. When this generalization is made it becomes possible to prove (generalized) completeness for the logic. However, if a tableau system is used it will contain a Cut rule. In order to avoid invoking the latter it seems to be necessary to adopt a second generalization and to move to a class of structures that do not necessarily validate the axiom of Extensionality, which says that two predicates are identical when they can be predicated of the same objects. The strategy of taking out Extensionality, pioneered by Takahashi [49] and Prawitz [45], allows one to prove the completeness of a cut-free system, after which Extensionality can be added to the logic again if that should be desired.

In the present set-up, which is inspired by Fitting [19], we will get rid of Extensionality by distinguishing between the *intension* and the *extension* of a term of complex type. The basic idea will be that any object in a domain D_α can be the intension of some term. Intensions of complex type will not be constructed set-theoretically out of those of a less complex type. Extensions, on the other hand, will be relations over the relevant domains of intensions, with their identity criteria therefore given by set membership. One and the same extension may be determined by two or more different intensions.

Let us see how this can be done. A *collection of domains* will be a set of non-empty sets $\{D_\alpha \mid \alpha \in \mathcal{T}\}$, such that $D_\alpha \cap D_\beta = \varnothing$ if $\alpha \neq \beta$. There are no further constraints

on collections of domains. *Assignments* and the notational conventions pertaining to assignments are defined as before. The set of all assignments for a collection of domains D is denoted $\mathcal{A}_D$. The *intension functions* defined below send terms to almost arbitrary domain elements. There are a few restrictions on these functions but they are rather liberal.

DEFINITION 13. An *intension function* for a collection of domains $D = \{D_\alpha \mid \alpha \in \mathcal{T}\}$ and a language $\mathcal{L}$ is a function I with domain $\mathcal{A}_D \times T^{\mathcal{L}}$ such that

(i) $I(a, A) \in D_\alpha$, if A is of type α

(ii) $I(a, x) = a(x)$, if x is a variable

(iii) $I(a, A) = I(a', A)$, if a and a' agree on all variables free in A

(iv) $I(a, A\{x := B\}) = I(a[I(a, B)/x], A)$, if B is free for x in A

Before we continue with also defining *extension* functions, let us pay some attention to the nitty-gritty and observe that the intension functions just defined behave well when the language is restricted or extended. The following property will be used a couple of times below.

PROPOSITION 14. *(i) Let I be an intension function for D and $\mathcal{L}$ and let $\mathcal{L}' \subseteq \mathcal{L}$. Then the restriction I' of I to $\mathcal{A}_D \times T^{\mathcal{L}'}$ is an intension function for D and $\mathcal{L}'$. (ii) Let I be an intension function for D and $\mathcal{L}$, let $\mathcal{L} \subseteq \mathcal{L}'$ and let f be a function with domain $\mathcal{L}' \backslash \mathcal{L}$ such that $f(c) \in D_\alpha$ if $c \in \mathcal{L}'_\alpha \backslash \mathcal{L}_\alpha$. Then there is an intension function I' for D and $\mathcal{L}'$ such that I and I' agree on $\mathcal{A}_D \times T^{\mathcal{L}}$ and $I'(a, c) = f(c)$ for every $c \in \mathcal{L}' \backslash \mathcal{L}$.*

Proof. (i) is trivial, so let us verify (ii). Let A be an arbitrary term in $T^{\mathcal{L}'}$ and let $c_1, \ldots, c_n$ be the constants occurring in A that are in $\mathcal{L}'$ but not in $\mathcal{L}$ such that $c_i <_{\mathcal{L}'} c_j$ if $i < j$. Let $A^\dagger$ be the result of replacing each c_i in A with the first variable x_i in $<_{\mathcal{V}}$ such that x_i is not free in A, has the type of c_i and is distinct from each of the x_j $(j < i)$. Clearly $A = A^\dagger\{x_1 := c_1, \ldots, x_n := c_n\}$. Let $I'(a, A) = I(a[f(c_1)/x_1, \ldots, f(c_n)/x_n], A^\dagger)$ and check that I' meets the requirements. ❑

The next definition provides the promised extension functions, which send objects of complex type to certain relations. We first give very general constraints; more requirements will follow in definition 17.

DEFINITION 15. An *extension function* for $D = \{D_\alpha \mid \alpha \in \mathcal{T}\}$ is a function E with domain $\cup\{D_\alpha \mid \alpha \text{ is complex}\}$ such that $E(d) \subseteq D_s \times D_{\alpha_1} \times \cdots \times D_{\alpha_n}$ if $d \in D_{\langle \alpha_1 \ldots \alpha_n \rangle}$.

Note that there is no requirement that the restriction of an extension function to any $D_{\langle \alpha_1 \ldots \alpha_n \rangle}$ should be *onto* $\mathcal{P}(D_s \times D_{\alpha_1} \times \cdots \times D_{\alpha_n})$ or that extension functions should be injective. This reflects the two generalizations discussed above. The possible lack of surjectivity is Henkin's generalization and the possible lack of injectivity reflects the move that Prawitz and Takahashi made.

DEFINITION 16. A *generalized frame* for the language $\mathcal{L}$ is a triple $\langle D, I, E \rangle$ such that D is a collection of domains, I is an intension function for D and $\mathcal{L}$, and E is an extension function for D.

We are interested in the extensions $E(I(a, A))$ of terms A. For the sake of readability we will often write $V(a, A)$ for these, letting V denote the composition of E and I. The following definition puts a series of constraints on extension functions that make things start to behave in a desired way.

DEFINITION 17. A generalized frame $\langle D, I, E\rangle$ for $\mathcal{L}$ is a *intensional model* for $\mathcal{L}$ if

(i) $V(a, AB) = \{\langle w, \vec{d}\rangle \mid \langle w, I(a, B), \vec{d}\rangle \in V(a, A)\}$;

(ii) $V(a, \lambda x_\beta.A) = \{\langle w, d, \vec{d}\rangle \mid d \in D_\beta \text{ and } \langle w, \vec{d}\rangle \in V(a[d/x], A)\}$;

(iii) $V(a, \bot) = \varnothing$;

(iv) $V(a, \varphi \to \psi) = D_s - (V(a, \varphi) - V(a, \psi))$;

(v) $V(a, \forall x_\alpha \varphi) = \bigcap_{d \in D_\alpha} V(a[d/x], \varphi)$;

(vi) $V(a, \langle R\rangle\varphi) = \{w \mid \exists w' \in V(a, \varphi) \text{ such that } \langle w, w'\rangle \in V(a, R)\}$;

(vii) $V(a, R^{\smile}) = \{\langle w, w'\rangle \mid \langle w', w\rangle \in V(a, R)\}$.

The clauses here are identical to the relevant ones in definition 12, with one important exception: in the clause for AB the (second slice function of) the value of A is no longer applied to the *extension* of B, but to its *intension*. The idea is that the extension of a predicate A determines and is determined by all the things that A can truthfully be predicated of while the intension of A determines and is determined by all the predicates that hold of A.

Do intensional models exist? One answer is that the standard models defined in the previous section obviously correspond to a subclass of the class of intensional models in which E is the identity function, but one would like to see intensional models that are not standard. For the latter we refer to the construction in the section on elementary model theory below.

Having the notion of intensional model in place we can define what it means for a sentence to be made true by an intensional model in a given world or to be valid in an intensional model.

DEFINITION 18. Let $M = \langle D, I, E\rangle$ be an intensional model for $\mathcal{L}$, let $w \in D_s$ and let φ be a sentence of $\mathcal{L}$. M and w *satisfy* φ (or *make* φ *true*), $M, w \models \varphi$, if $w \in V(a, \varphi)$ for any a. We also say that M satisfies φ if there is some $w \in D_s$ such that $M, w \models \varphi$ and that φ is *satisfiable* if some M satisfies φ. If $M, w \models \varphi$ for all $w \in D_s$ then φ is said to be *valid* in M and we write $M \models \varphi$.

The corresponding notion of entailment is defined as follows.

DEFINITION 19. Let Π and Σ be sets of sentences in $\mathcal{L}$. Π is said to *intensionally entail* or *i-entail* Σ, $\Pi \models_i \Sigma$, if, for every intensional model $M = \langle D, I, E\rangle$ for $\mathcal{L}$ and every $w \in D_s$, if $M, w \models \varphi$ for all $\varphi \in \Pi$ then $M, w \models \varphi$ for some $\varphi \in \Sigma$.

This gives a rather weak logic in comparison with other type logics. In applications it will usually be necessary to strengthen the logic with sets of sentences $\mathcal{S}$ which may typically contain modal axioms, but may also contain classical axioms, such as instantiations of the Extensionality scheme, the Axiom of Descriptions, or axioms regulating λ-conversion. About the latter notion the following proposition lists some useful facts.

PROPOSITION 20. *Let $M = \langle D, I, E\rangle$ be an intensional model, and let a be an assignment for D. Then, for all A and B of appropriate types,*

(i) $V(a, \lambda x.A) = V(a, \lambda y.A\{x := y\})$, *if y is free for x in A;*

(ii) $V(a, (\lambda x.A)B) = V(a, A\{x := B\})$, *if B is free for x in A;*

(iii) $V(a, \lambda x.Ax) = V(a, A)$, *if x is not free in A.*

Proof. Left to the reader. ❑

$$\frac{\Gamma}{\Gamma'}\,\mathrm{W} \qquad \text{if } \Gamma' \subseteq \Gamma$$

$$\frac{\Gamma, \mathsf{T}u{:}\,\bot}{}\,\mathsf{T}\bot \qquad\qquad \frac{\Gamma, \mathsf{T}u{:}\,\varphi,\ \mathsf{F}u{:}\,\varphi}{}\,\mathrm{Ax}$$

$$\frac{\Gamma, \mathsf{S}u{:}\,(\lambda x.A)B\vec{C}}{\Gamma, \mathsf{S}u{:}\,A\{x := B\}\vec{C}}\,\beta\text{-ext} \qquad\qquad \frac{\Gamma, \mathsf{S}u{:}\,R^{\smile}u'}{\Gamma, \mathsf{S}u'{:}\,Ru}\,\smile$$

$$\frac{\Gamma, \mathsf{T}u{:}\,\varphi \to \psi}{\Gamma, \mathsf{F}u{:}\,\varphi \,\big|\, \Gamma, \mathsf{T}u{:}\,\psi}\,\mathsf{T}{\to} \qquad\qquad \frac{\Gamma, \mathsf{F}u{:}\,\varphi \to \psi}{\Gamma, \mathsf{T}u{:}\,\varphi,\ \mathsf{F}u{:}\,\psi}\,\mathsf{F}{\to}$$

$$\frac{\Gamma, \mathsf{T}u{:}\,\forall x\varphi}{\Gamma, \mathsf{T}u{:}\,\varphi\{x := A\}}\,\mathsf{T}\forall \qquad\qquad \frac{\Gamma, \mathsf{F}u{:}\,\forall x\varphi}{\Gamma, \mathsf{F}u{:}\,\varphi\{x := c\}}\,\mathsf{F}\forall \ \ (c \text{ not in the premise})$$

$$\frac{\Gamma, \mathsf{T}u{:}\,\langle R\rangle\varphi}{\Gamma, \mathsf{T}u{:}\,Ru', \mathsf{T}u'{:}\,\varphi,}\,\mathsf{T}\langle\cdot\rangle \ \ (u' \text{ not in the premise}) \qquad\qquad \frac{\Gamma, \mathsf{F}u{:}\,\langle R\rangle\varphi}{\Gamma, \mathsf{F}u{:}\,Ru' \,\big|\, \Gamma, \mathsf{F}u'{:}\,\varphi}\,\mathsf{F}\langle\cdot\rangle$$

Table 1. Tableau rules for **MTT**.

These statements show that λ-conversion preserves identity of extension, but that does not imply that intensional identity is also preserved and that V can be replaced uniformly with I in the proposition above. If such intensional identities are wanted, and in most applications one will certainly want to have at least the possibility of α and β conversion in any context, an axiomatic extension of the logic may provide them. See 5.2 below.

5 TABLEAUS FOR MODAL TYPE THEORY

5.1 Tableaus

In this section the proof theory of **MTT** will be given in the form of a tableau system. The calculus will be set up as a form of labeled deduction, with labels storing information about worlds and truth values. Formally, a *labeled sentence* of $\mathcal{L}$ will be a triple $\langle \mathsf{S}, u, \varphi\rangle$ consisting of a *sign* S, which can either be T or F, a constant $u \in \mathcal{L}_s$, and a sentence φ of $\mathcal{L}$. Labeled sentences $\langle \mathsf{S}, u, \varphi\rangle$ will typically be written as $\mathsf{S}u{:}\,\varphi$, where $\mathsf{T}u{:}\,\varphi$ can be read as 'φ is true in world u' and $\mathsf{F}u{:}\,\varphi$ as expressing that φ is false in u.

Tableaus will be defined as certain sets of branches. A *branch* in its turn will be a set of labeled sentences. The notion of *satisfaction* can easily be extended from sentences to labeled sentences and branches, for we can define an intensional model $M = \langle D, I, E\rangle$ to satisfy $\mathsf{T}u{:}\,\varphi$ if $I(a, u) \in V(a, \varphi)$ for some (and hence every) a, while letting M satisfy $\mathsf{F}u{:}\,\varphi$ if $I(a, u) \notin V(a, \varphi)$ for any a. M is said to *satisfy* a branch Γ if it satisfies all $\vartheta \in \Gamma$. If no model M satisfies Γ, Γ is said to be *unsatisfiable*; otherwise Γ is *satisfiable*.

We will use the usual sequent notation for branches, writing Γ, θ for $\Gamma \cup \{\theta\}$, etc. Diverging slightly from the usual set-up of tableaus, tableau rules will be defined as certain relations between branches, not as relations between labeled sentences. The interpretation of these rules (that are given in Table 1) is one of replacement of branches,

$$\frac{\mathsf{T}u\colon \neg\varphi}{\mathsf{F}u\colon \varphi}\,\mathsf{T}\neg \qquad \frac{\mathsf{F}u\colon \neg\varphi}{\mathsf{T}u\colon \varphi}\,\mathsf{F}\neg \qquad \frac{\mathsf{T}u\colon \varphi\wedge\psi}{\mathsf{T}u\colon \varphi,\ \mathsf{T}u\colon \psi}\,\mathsf{T}\wedge \qquad \frac{\mathsf{F}u\colon \varphi\wedge\psi}{\mathsf{F}u\colon \varphi \mid \mathsf{F}u\colon \psi}\,\mathsf{F}\wedge$$

$$\frac{\mathsf{T}u\colon \varphi\vee\psi}{\mathsf{T}u\colon \varphi \mid \mathsf{T}u\colon \psi}\,\mathsf{T}\vee \qquad \frac{\mathsf{F}u\colon \varphi\vee\psi}{\mathsf{F}u\colon \varphi,\ \mathsf{F}u\colon \psi}\,\mathsf{F}\vee \qquad \frac{\mathsf{T}u\colon \exists x\varphi}{\mathsf{T}u\colon \varphi\{x:=c\}}\,\mathsf{T}\exists\ (c \text{ fresh}) \qquad \frac{\mathsf{F}u\colon \exists x\varphi}{\mathsf{F}u\colon \varphi\{x:=A\}}\,\mathsf{F}\exists$$

$$\frac{\mathsf{T}u\colon \varphi\leftrightarrow\psi}{\mathsf{T}u\colon \varphi,\ \mathsf{T}u\colon \psi \mid \mathsf{F}u\colon \varphi,\ \mathsf{F}u\colon \psi}\,\mathsf{T}\leftrightarrow \qquad \frac{\mathsf{F}u\colon \varphi\leftrightarrow\psi}{\mathsf{T}u\colon \varphi,\ \mathsf{F}u\colon \psi \mid \mathsf{F}u\colon \varphi,\ \mathsf{T}u\colon \psi}\,\mathsf{F}\leftrightarrow$$

$$\frac{\mathsf{T}u\colon [R]\varphi}{\mathsf{F}u\colon Ru' \mid \mathsf{T}u'\colon \varphi}\,\mathsf{T}[\cdot] \qquad \frac{\mathsf{F}u\colon [R]\varphi}{\mathsf{T}u\colon Ru', \mathsf{F}u'\colon \varphi}\,\mathsf{F}[\cdot]\ (u' \text{ fresh}) \qquad \frac{\mathsf{T}u'\colon \dot{u}}{\mathsf{T}u'\colon u=u'}\,\mathsf{T}^{\cdot} \qquad \frac{\mathsf{F}u\colon \dot{u}}{}\,\mathsf{F}^{\cdot}$$

$$\frac{\mathsf{T}u\colon \Box\varphi}{\mathsf{T}u'\colon \varphi}\,\mathsf{T}\Box \qquad \frac{\mathsf{F}u\colon \Box\varphi}{\mathsf{F}u'\colon \varphi}\,\mathsf{F}\Box\ (u' \text{ fresh}) \qquad \frac{\mathsf{T}u\colon \Diamond\varphi}{\mathsf{T}u'\colon \varphi}\,\mathsf{T}\Diamond\ (u' \text{ fresh}) \qquad \frac{\mathsf{F}u\colon \Diamond\varphi}{\mathsf{F}u'\colon \varphi}\,\mathsf{F}\Diamond$$

$$\frac{\mathsf{T}u\colon A=B,\ \mathsf{S}u\colon \varphi\{x:=A\}}{\mathsf{S}u\colon \varphi\{x:=B\}}\,\mathrm{LL} \qquad \frac{}{\mathsf{T}u\colon A=A}\,\mathrm{id} \qquad \frac{\mathsf{T}u\colon A=B}{\mathsf{T}u'\colon A=B}\,\mathrm{U}$$

$$\frac{\mathsf{S}u\colon \varphi,\ \mathsf{T}u\colon u=u'}{\mathsf{S}u'\colon \varphi}\,\mathrm{LL}' \qquad \frac{}{\mathsf{T}u\colon \varphi \mid \mathsf{F}u\colon \varphi}\,\mathrm{Cut}$$

Table 2. Derived tableau rules and Cut (abbreviated forms).

for example the interpretation of $\mathsf{T}\!\to$ in Table 1 is that the branch $\Gamma, \mathsf{T}u\colon \varphi \to \psi$ can be replaced by the two branches $\Gamma, \mathsf{F}u\colon \varphi$ and $\Gamma, \mathsf{T}u\colon \psi$ in any tableau. The format also allows the formulation of a *weakening* rule W that allows the removal of signed formulas from a branch.

Compare $\mathsf{T}\!\to$ with a more usual approach where one would have a rule

$$\frac{\mathsf{T}u\colon \varphi\to\psi}{\mathsf{F}u\colon \varphi \mid \mathsf{T}u\colon \psi}\,,$$

meaning that whenever a branch is found to contain $\mathsf{T}u\colon \varphi \to \psi$ it may be split, $\mathsf{F}u\colon \varphi$ may be added to one side and $\mathsf{T}u\colon \psi$ to another. Of course the two approaches very much boil down to the same thing. The present set-up is close to that of a Gentzen calculus for the logic: read T as 'left' and F as 'right' and turn the rules in Table 1 upside down.

A convention that is adopted in Table 1 (and that we shall continue to use) is that wherever the notation $A\{x := B\}$ is used B must be free for x in A. An alternative notation for tableau rules, better suited for inline environments, is $\Gamma/\Gamma_1;\ldots;\Gamma_n$, where / replaces the horizontal line and ; the vertical lines in any rule. The following definition tells how we can expand sets of branches and obtain tableaus.

DEFINITION 21. A set of branches T' is a *one step expansion* of a set of branches T if $T' = (T\backslash\Gamma) \cup \{\Gamma_1,\ldots,\Gamma_n\}$ for some tableau rule $\Gamma/\Gamma_1;\ldots;\Gamma_n$. T' is an *expansion* of T if there is a sequence $T_1,\ldots,T_n$ such that $T_1 = T$, $T_n = T'$ and each T_{k+1} is a one step

Name	Modal axiom	Corresponding R rule
$\mathbf{T}_\forall$	$\Box\forall p([R]p \to p)$	$/\mathsf{T}u{:}\, Ru$
$\mathbf{D}_\forall$	$\Box\forall p([R]p \to \langle R\rangle p)$	$/\mathsf{T}u_1{:}\, Ru_2$ (u_2 fresh)
$\mathbf{4}_\forall$	$\Box\forall p([R]p \to [R][R]p)$	$\mathsf{T}u_1{:}\, Ru_2, \mathsf{T}u_2{:}\, Ru_3/\mathsf{T}u_1{:}\, Ru_3$
$\mathbf{5}_\forall$	$\Box\forall p(\langle R\rangle p \to [R]\langle R\rangle p)$	$\mathsf{T}u_1{:}\, Ru_2, \mathsf{T}u_1{:}\, Ru_3/\mathsf{T}u_2{:}\, Ru_3$

Table 3. Correspondences between modal axioms and certain rules.

expansion of T_k. A set of branches T is a *tableau* if it is an expansion of $\{\Gamma\}$ for some finite branch Γ.

Thus while no finiteness condition was imposed on branches per se, tableaus are stipulated to originate from finite branches. Note that the $\mathsf{T}\bot$ and Ax rules can cause branches to disappear from a tableau while it is being expanded. This can lead to the closure of tableaus as defined in the following definition.

DEFINITION 22. A finite branch Γ *has a closed tableau* if $\varnothing$ is an expansion of $\{\Gamma\}$. If Π and Σ are sets of sentences then $\Pi \vdash \Sigma$ holds if, for some finite $\Pi_0 \subseteq \Pi$, some finite $\Sigma_0 \subseteq \Sigma$ and some $u \in \mathcal{L}_s$ that does not occur in any sentence in $\Pi_0 \cup \Sigma_0$, $\{\mathsf{T}u{:}\,\varphi \mid \varphi \in \Pi_0\} \cup \{\mathsf{F}u{:}\,\varphi \mid \varphi \in \Sigma_0\}$ has a closed tableau.

We employ the usual notational conventions with respect to $\vdash$. A formula φ is called *tableau provable* if $\vdash \varphi$.

For ease of reference Table 2 lists some rules that are derivable from those already given in Table 1. We leave it to the reader to show that these rules are indeed derivable (most cases are entirely trivial, some easy but amusing). Another exercise is to show that $\vdash \exists p(p \wedge \forall q(q \to [R](p \to q)))$. Table 2 also displays the Cut rule, which we will see is admissable. Here we have not bothered to write all the Γs of our official rule presentation and have reverted to the more usual way of presenting tableau rules.

Clearly the rules were chosen in a way that makes it possible to show Soundness to hold.

THEOREM 23 (Soundness). *If Γ has a closed tableau then Γ is unsatisfiable. Hence $\Pi \vdash \Sigma$ implies $\Pi \models_i \Sigma$.*

Proof. For each tableau rule $\Gamma/\Gamma_1; \ldots; \Gamma_n$, if Γ is satisfiable, one of the Γ_i is satisfiable. Verifying this will involve proposition 14 for some cases. By induction, if T is an expansion of $\{\Gamma\}$ then, if Γ is satisfiable, some $\Gamma' \in T$ must be satisfiable. Hence if Γ has a closed tableau, Γ can not be satisfiable. This proves the first statement of the theorem. Suppose $\Pi \vdash \Sigma$. Then for some finite $\Pi_0 \subseteq \Pi$ and $\Sigma_0 \subseteq \Sigma$ and some $u \in \mathcal{L}_s$ that does not occur in $\Pi_0 \cup \Sigma_0$, $\{\mathsf{T}u{:}\,\varphi \mid \varphi \in \Pi_0\} \cup \{\mathsf{F}u{:}\,\varphi \mid \varphi \in \Sigma_0)\}$ has a closed tableau and hence is unsatisfiable. It follows that $\Pi \models_i \Sigma$ ❑

5.2 Axiomatic Extensions

If, in some setting, one wants to restrict attention to a class of models that validate some set of sentences $\mathcal{S}$ then it becomes natural to define $\Pi \models_{\mathcal{S}} \Sigma$ as $\mathcal{S} \cup \Pi \models_i \Sigma$. Similarly, $\Pi \vdash_{\mathcal{S}} \Sigma$ can be defined as $\mathcal{S} \cup \Pi \vdash \Sigma$ and the soundness theorem gives that $\Pi \vdash_{\mathcal{S}} \Sigma$

$$\mathsf{T}u\colon \Box\forall p([R]p \to [R][R]p)$$
$$\mathsf{T}u_1\colon Ru_2$$
$$\mathsf{T}u_2\colon Ru_3$$
$$\mathsf{T}u_1\colon \forall p([R]p \to [R][R]p)$$
$$\mathsf{T}u_1\colon [R]R^{\smile}u_1 \to [R][R]R^{\smile}u_1$$

$$\begin{array}{c|c}
\begin{array}{c}
\mathsf{F}u_1\colon [R]R^{\smile}u_1\\
\mathsf{T}u_1\colon Ru'\\
\mathsf{F}u'\colon R^{\smile}u_1\\
\mathsf{F}u_1\colon Ru'\\
\times
\end{array}
&
\begin{array}{c}
\mathsf{T}u_1\colon [R][R]R^{\smile}u_1\\
\begin{array}{c|c}
\begin{array}{c}\mathsf{F}u_1\colon Ru_2\\ \times\end{array}
&
\begin{array}{c}
\mathsf{T}u_2\colon [R]R^{\smile}u_1\\
\begin{array}{c|c}
\begin{array}{c}\mathsf{F}u_2\colon Ru_3\\ \times\end{array}
&
\begin{array}{c}\mathsf{T}u_3\colon R^{\smile}u_1\\ \mathsf{T}u_1\colon Ru_3\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}$$

Figure 1. Derivation of $\mathsf{T}u_1\colon Ru_2, \mathsf{T}u_2\colon Ru_3/\mathsf{T}u_1\colon Ru_3$ from $\mathbf{4}_\forall$

implies $\Pi \models_{\mathcal{S}} \Sigma$ (while completeness, yet to be shown to hold, gives the converse). Prime candidates for inclusion in such a theory $\mathcal{S}$ are the usual rules for lambda conversion. These are the universal closures of any instantiation of one of the following schemes.

(α) $\lambda x.A = \lambda y.A\{x := y\}$, if y is free for x in A;

(β) $(\lambda x.A)B = A\{x := B\}$, if B is free for x in A;

(η) $\lambda x.Ax = A$, if x is not free in A.

It is clearly consistent to add these rules to **MTT**, as they are valid in standard models. Once they are added, the derived rules U and LL ensure that

$$\frac{\mathsf{S}u\colon \varphi\{x := A\}}{\mathsf{S}u\colon \varphi\{x := B\}}\,\lambda \qquad \text{if } A =_{\beta\eta} B$$

also becomes a derived rule. Since one can work with the standard notion of reduction $\twoheadrightarrow_{\beta\eta}$ here, it is clear that the resulting logic is Church-Rosser. This will also hold if, for some reason, it is decided that $\mathcal{S}$ should contain (α), but only one of the rules (β) and (η). Note that the (β) rule scheme discussed here should well be distinguished from the rule we have called β-ext, which is much weaker, as it only allows β-conversions in *head* position.

Other obvious candidates for inclusion in a theory $\mathcal{S}$ are the usual modal axioms for modalities $\langle R\rangle$. For instance, one could ensure validity of $\mathbf{T}$ by including the scheme $\Box([R]\varphi \to \varphi)$ (the leading $\Box$ ensures that one gets validity, not just truth of $\mathbf{T}$ in the initial world). Another way to express the same idea, more natural perhaps in the present context, is by quantification over propositions, as in $\Box\forall p([R]p \to p)$, which is called $\mathbf{T}_\forall$ in Table 3, where also quantified analogues of $\mathbf{D}$, $\mathbf{4}$ and $\mathbf{5}$ are found. If such axioms are adopted it is often possible to use derived rules in one's tableaus that closely mirror the usual frame correspondences in modal logic (for the latter see e.g. Chapter 1 of this handbook, by Blackburn and Van Benthem). In fig. 1, for example, is a tableau verifying that $\mathsf{T}u_1\colon Ru_2, \mathsf{T}u_2\colon Ru_3/\mathsf{T}u_1\colon Ru_3$ is a derived rule in the presence of the $\mathbf{4}_\forall$ axiom. On the other hand, if $\mathsf{T}u_1\colon Ru_2, \mathsf{T}u_2\colon Ru_3/\mathsf{T}u_1\colon Ru_3$ should be adopted as an additional rule, $\Box\forall p[R]p \to [R][R]p$ becomes tableau provable, as fig. 2 shows. Table 3 lists some more of

$$
\begin{array}{c}
\mathsf{F}u\colon \Box\forall p([R]p \to [R][R]p)\\
\mathsf{F}u_1\colon \forall p([R]p \to [R][R]p)\\
\mathsf{F}u_1\colon [R]c_{\langle\rangle} \to [R][R]c\\
\mathsf{T}u_1\colon [R]c\\
\mathsf{F}u_1\colon [R][R]c\\
\mathsf{T}u_1\colon Ru_2\\
\mathsf{F}u_2\colon [R]c\\
\mathsf{T}u_2\colon Ru_3\\
\mathsf{F}u_3\colon c\\
\mathsf{T}u_1\colon Ru_3\\
\begin{array}{cc}
\mathsf{F}u_1\colon Ru_3 & \mathsf{T}u_3\colon c\\
\times & \times
\end{array}
\end{array}
$$

Figure 2. Derivation of $\mathbf{4}_\forall$ in the presence of $\mathsf{T}u_1\colon Ru_2, \mathsf{T}u_2\colon Ru_3/\mathsf{T}u_1\colon Ru_3$

these correspondences (use a nominal $\dot{u}_3$ when showing the correctness of the last one) and the reader will have no difficulty providing many more. Note that, with the help of nominals, it is also possible to directly express properties of accessibility relations, even those that are not modally definable in the usual set-up. For example, irreflexivity of R can be expressed as $\Box\forall x_s(\dot{x} \to \neg\langle R\rangle\dot{x})$. See Blackburn et al. [10] for more information on expressing first order relational properties with the help of nominals.

6 ELEMENTARY MODEL THEORY

In this section we will prove some basic modeltheoretic properties of **MTT**: Generalized Completeness, the Generalized Löwenheim-Skolem property, and the admissability of the Cut rule, all via a Model Existence theorem in the way Smullyan [48] did it for first order logic (see also Fitting [17, 19]). None of the techniques employed here is new, but we include full proofs for two reasons. The first of these being that, since our definition of an intensional model deviates from existing notions in the literature and since the devil is always in the details, it is good to have an explicit sanity check on those definitions. The second reason is that readers not already familiar with these kind of proofs may find examples here in a relatively streamlined setting.

Before we tackle the main modeltheoretic properties of **MTT**, some attention must be paid to the notion of *identity* in intensional models, as this relation may not be the identity of the metalanguage.

6.1 Identity and Indiscernability

The decision to let the relations $=_{\langle\alpha\alpha\rangle}$ be abbreviations of $\lambda x_\alpha\lambda y_\alpha\forall z_{\langle\alpha\rangle}(zx \to zy)$, as it was done in definition 9, derives directly from Russell, and via Russell from Leibniz, as the abbreviation equates identity with *indistinguishability*. It is clear that in standard

models identity and indistinguishability coincide, but, as was noted by Andrews [2] for the non-modal case, in nonstandard models it may happen that two objects d_1 and d_2 that are in fact not identical may fail to be distinguished because there simply is no set to keep them apart. This may be thought of as an anomaly and one may be tempted to restrict attention to intensional models that are *normal* in the following sense.

DEFINITION 24. An intensional model $M = \langle D, I, E\rangle$ is *normal* if, for any α, any $d, d' \in D_\alpha$ and arbitrary a, $d = d'$ if $\langle w, d, d'\rangle \in V(a, =)$ for some $w \in D_s$.

In fact restriction to normal models will not buy us any new truths as will be shown shortly. First some facts that will come in handy.

PROPOSITION 25. *Let $M = \langle D, I, E\rangle$ be an intensional model, and let a be an assignment for D. Then, for all A, B and B' of appropriate types,*

(i) $V(a, A = B) = \varnothing$ *or* $V(a, A = B) = D_s$*;*

(ii) $V(a, A = A) = D_s$*;*

(iii) $V(a, A\{x := B\} = A\{x := B'\}) = D_s$ *if* $V(a, B = B') = D_s$*, provided B and B' are free for x in A.*

Proof. (i) Suppose $w \in V(a, A_\alpha = B_\alpha)$, i.e. $w \in V(a, \forall x(xA \to xB))$. Choosing $\lambda y_\alpha.\Box\forall z_{\langle\alpha\rangle}(zA \to zy)$ for x, it is easily shown that $w \in V(a, \Box\forall z(zA \to zB))$. Hence $w' \in V(a, \forall x(xA \to xB))$ for all $w' \in D_s$ and we are done. (ii) Trivial. (iii) Assume that $w \in V(a, B = B'$, i.e. $w \in V(a, \forall y(yB \to yB'))$. Choose $\lambda v.A\{x := B\} = A\{x := v\}$ (with fresh v) for y and derive that $w \in V(a, A\{x := B\} = A\{x := B'\})$. ❑

The following proposition shows that, if desired, one can always 'normalize' models by 'dividing out' the indistinguishability relation. The proof implicitly uses the axiom of choice.

PROPOSITION 26. *Let $M = \langle D, I, E\rangle$ be an intensional model and let $w \in D_s$. There are a normal intensional model $\overline{M} = \langle \overline{D}, \overline{I}, \overline{E}\rangle$ and a $\overline{w} \in \overline{D}_s$ such that, for each sentence φ, $\overline{w}$ satisfies φ in $\overline{M}$ iff w satisfies φ in M.*

Proof. Suppose $M = \langle D, I, E\rangle$. We define the relation $\sim$ between objects of identical type in M's domains as follows. For any α, any $d, d' \in D_\alpha$ and arbitrary a let $d \sim d'$ iff, for some (and therefore every) $w \in D_s$, $\langle w, d, d'\rangle \in V(a, =_{\langle\alpha\alpha\rangle})$. Clearly, $\sim$ is an equivalence relation. Using proposition 25 and definition 13 it is straightforward to show that, for any term A,

$$(16)\quad d \sim d' \Longrightarrow I(a[d/x], A) \sim I(a[d'/x], A)\ .$$

It is also worth noting that, for any w, w' and any φ and a

$$(17)\quad w \sim w' \Longrightarrow (w \in V(a, \varphi) \Longrightarrow w' \in V(a, \varphi))\ .$$

The way to show this is to observe that, if neither x_s nor y_s is free in φ,

$$(18)\quad V(a[w/x], (\lambda y.\varphi)^{\smile}\, x) = D_s \Longleftrightarrow w \in V(a, \varphi)\ ,$$

and to then use the definition of $w \sim w'$.

Define $\overline{d} = \{d' \mid d \sim d'\}$, and let $\overline{D}_\alpha = \{\overline{d} \mid d \in D_\alpha\}$, while $\overline{D} = \{\overline{D}_\alpha \mid \alpha \in \mathcal{T}\}$. Let f be a function such that $f(\overline{d}) \in \overline{d}$, if $\overline{d} \in \overline{D}_\alpha$. For any assignment a for $\overline{D}$, let a°

be the assignment for D defined by $a^\circ(x) = f(a(x))$, for all x. Let $\overline{I}(a, A) = \overline{I(a^\circ, A)}$, for each assignment a for $\overline{D}$ and each term A. Then $\overline{I}$ is an intension function for $\overline{D}$. The first three requirements of definition 13 are easily checked, so let us check the last requirement. Note that

$$\begin{array}{rcl} I(a^\circ, A\{x := B\}) & = & \text{(by definition 13)} \\ I(a^\circ[I(a^\circ, B)/x], A) & \sim & \text{(by (16))} \\ I(a^\circ[f(\overline{I(a^\circ, B)})/x]), A) & = & \text{(by the definition of } \overline{I}) \\ I(a^\circ[f(\overline{I}(a, B))/x]), A) & = & \text{(by the definition of } \circ) \\ I((a[\overline{I}(a, B)/x])^\circ, A)\ . & & \end{array}$$

From this conclude that $\overline{I}(a, A\{x := B\}) = \overline{I}(a[\overline{I}(a, B)/x], A)$.

Define $\overline{E}$ by letting $\overline{E}(\overline{d_\alpha}) = \{\langle \overline{w}, \overline{d_1}, \ldots, \overline{d_n}\rangle \mid \langle w, d_1, \ldots, d_n\rangle \in E(d)\}$, if α is complex. In order to show that this is well-defined assume that $w \sim w'$, $d \sim d'$, and $d_i \sim d'_i$. Then, if $R, x_1, \ldots, x_n$ are distinct variables of appropriate types

$$\begin{array}{rcl} \langle w, d_1, \ldots, d_n\rangle \in E(d) & \Longleftrightarrow & \text{(by def. (17))} \\ w \in V(a[d/R, d_1/x_1, \ldots, d_n/x_n], Rx_1 \ldots x_n) & \Longleftrightarrow & \text{(by (17))} \\ w' \in V(a[d/R, d_1/x_1, \ldots, d_n/x_n], Rx_1 \ldots x_n) & \Longleftrightarrow & \text{(equational reas.)} \\ w' \in V(a[d'/R, d'_1/x_1, \ldots, d'_n/x_n], Rx_1 \ldots x_n) & \Longleftrightarrow & \text{(by def. (17))} \\ \langle w', d'_1, \ldots, d'_n\rangle \in E(d') & & \end{array}$$

so that the definition was legitimate.

Write $\overline{V}(a, A)$ for $\overline{E}(\overline{I}(a, A))$ and observe that

(19) $\langle \overline{w}, \overline{d_1}, \ldots, \overline{d_n}\rangle \in \overline{V}(a, A)$ iff $\langle w, d_1, \ldots, d_n\rangle \in V(a^\circ, A)$

From this it follows by a straightforward induction on term complexity that $\overline{M} = \langle \overline{D}, \overline{I}, \overline{E}\rangle$ is a intensional model. It also follows that $\overline{w}$ satisfies φ in $\overline{M}$ iff w satisfies φ in M and that $\overline{M}$ is normal. ❑

So the notion of identity of the logic may diverge from the notion of identity employed in the metalanguage, but for notions like satisfiability and entailment this does not matter.

6.2 Model Existence

We now come to the Model Existence theorem and its proof. This theorem (Theorem 30 below) roughly says that if a branch Γ does not have a certain kind of property, here called a 'sound unsatisfiability property', it is satisfiable by an intensional model. The proof proceeds in two steps. One step shows that such a branch Γ can be extended to a branch Γ^* that is *downward saturated* in a sense to be defined shortly. The other step shows that downward saturated branches are satisfiable. We will prove the last of these two statements first. Let us start with the definition of a downward saturated branch.

DEFINITION 27. A branch Γ of $\mathcal{L}$ is called *downward saturated* if the following hold:

(a) $\{\mathsf{T}u\colon \varphi,\ \mathsf{F}u\colon \varphi\} \not\subseteq \Gamma$ for any sentence φ and constant u;

(b) $\mathsf{T}u\colon \bot \notin \Gamma$;

(c) $\mathsf{S}u\colon (\lambda x.A)B\vec{C} \in \Gamma \Longrightarrow \mathsf{S}u\colon A\{x := B\}\vec{C} \in \Gamma$, if $\lambda x.A$, B, and the sequence of terms $\vec{C}$ are closed and of appropriate types;

(d) $\mathsf{T}u\colon \varphi \to \psi \in \Gamma \Longrightarrow \mathsf{F}u\colon \varphi \in \Gamma$ or $\mathsf{T}u\colon \psi \in \Gamma$;

(e) $\mathsf{F}u\colon \varphi \to \psi \in \Gamma \Longrightarrow \{\mathsf{T}u\colon \varphi,\ \mathsf{F}u\colon \psi\} \subseteq \Gamma$;

(f) $\mathsf{T}u\colon \forall x_\alpha \varphi \in \Gamma \Longrightarrow \mathsf{T}u\colon \varphi\{x := A\} \in \Gamma$ for all closed A of type α;

(g) $\mathsf{F}u\colon \forall x_\alpha \varphi \in \Gamma \Longrightarrow \mathsf{F}u\colon \varphi\{x := c\} \in \Gamma$ for some $c \in \mathcal{L}_\alpha$

(h) $\mathsf{T}u\colon \langle R\rangle\varphi \in \Gamma \Longrightarrow \{\mathsf{T}u\colon Ru', \mathsf{T}u'\colon \varphi\} \subseteq \Gamma$ for some $u' \in \mathcal{L}_s$;

(i) $\mathsf{F}u\colon \langle R\rangle\varphi \in \Gamma \Longrightarrow \mathsf{F}u\colon Ru' \in \Gamma$ or $\mathsf{F}u'\colon \varphi \in \Gamma$ for all $u' \in \mathcal{L}_s$;

(j) $\mathsf{S}u\colon R^{\smile}u' \in \Gamma \Longrightarrow \mathsf{S}u'\colon Ru \in \Gamma$;

A downward saturated branch Γ of $\mathcal{L}$ is said to be *complete* if $\mathsf{T}u\colon \varphi \in \Gamma$ or $\mathsf{F}u\colon \varphi \in \Gamma$ for each sentence φ of $\mathcal{L}$ and each $u \in \mathcal{L}_s$.

That downward saturated branches are satisfiable is the content of the next lemma. Here an intensional model is constructed using the method employed by Takahashi and Prawitz.

LEMMA 28 (Hintikka Lemma). *If Γ is a downward saturated branch in a language $\mathcal{L}$ such that $\mathcal{L}_\alpha \neq \varnothing$ for each basic type α then Γ is satisfied by a intensional model. If Γ is complete, then Γ is satisfied by a normal countable intensional model.*

Proof. Let Γ be a downward saturated branch in a language $\mathcal{L}$ as indicated. We will find an intensional model satisfying Γ using the Takahashi-Prawitz construction. The following induction on type complexity defines domains D_α as sets of pairs $\langle A, e\rangle$, where A is a closed term of type α and e is called a *possible extension* of A.

(i) If α is basic let $D_\alpha = \{\langle c, c\rangle \mid c \in \mathcal{L}_\alpha\}$;

(ii) If $\alpha = \langle \alpha_1 \dots \alpha_n\rangle$ let $\langle A_\alpha, e\rangle \in D_\alpha$ iff A is closed, $e \subseteq D_s \times D_{\alpha_1} \times \cdots \times D_{\alpha_n}$ and, whenever $\langle B_1, e_1\rangle \in D_{\alpha_1}, \dots, \langle B_n, e_n\rangle \in D_{\alpha_n}$

  (a) If $\mathsf{T}u\colon AB_1 \dots B_n \in \Gamma$ then $\langle\langle u, u\rangle, \langle B_1, e_1\rangle, \dots, \langle B_n, e_n\rangle\rangle \in e$;

  (b) If $\mathsf{F}u\colon AB_1 \dots B_n \in \Gamma$ then $\langle\langle u, u\rangle, \langle B_1, e_1\rangle, \dots, \langle B_n, e_n\rangle\rangle \notin e$.

Each D_α is non-empty. For basic types α this follows from the requirement that $\mathcal{L}_\alpha \neq \varnothing$; for complex types $\langle \alpha_1 \dots \alpha_n\rangle$ consider $\langle \lambda x_{\alpha_1} \dots \lambda x_{\alpha_n}.\bot, \varnothing\rangle$. Since induction on term complexity easily shows that terms have unique types, we also have that $D_\alpha \cap D_\beta = \varnothing$ if $\alpha \neq \beta$. Hence $\{D_\alpha \mid \alpha \in \mathcal{T}\}$ is a collection of domains. It is worth observing that each D_α is a function if Γ is complete. In that case each D_α will be countable.

The set $D = \{D_\alpha \mid \alpha \in \mathcal{T}\}$ will be the collection of domains of the intensional model we are after. We will define a function I which will turn out to be an intension function for D. First some handy notation. If π is an ordered pair, write π^1 and π^2 for the first and second elements of π respectively, so that $\pi = \langle \pi^1, \pi^2\rangle$. If f is a function whose values are ordered pairs, write f^1 and f^2 for the functions with the same domain as f, such that $f^1(z) = (f(z))^1$ and $f^2(z) = (f(z))^2$ for any argument z. Let a be an assignment for D. The substitution $\overleftarrow{a}$ is defined by $\overleftarrow{a}(x) = a^1(x)$ and we let $I^1(a, A) = A\overleftarrow{a}$ for any term A. The second component of I is defined using an induction on term complexity.

(a) $I^2(a, x) = a^2(x)$, if x is a variable;
$I^2(a, c_\alpha) = c$, if α is basic;
$I^2(a, c_\alpha) = \{\langle\langle u, u\rangle, \langle A_1, e_1\rangle, \dots, \langle A_n, e_n\rangle\rangle \mid \langle A_i, e_i\rangle \in D_{\alpha_i}$
$\& \ \mathsf{T}u\colon cA_1 \dots A_n \in \Gamma\}$, if α is complex;

(b) $I^2(a, AB) = \{\langle w, \vec{d}\rangle \mid \langle w, I(a,B), \vec{d}\rangle \in I^2(a,A)\}$;

(c) $I^2(a, \lambda x_\beta.A) = \{\langle w, d, \vec{d}\rangle \mid d \in D_\beta \text{ and } \langle w, \vec{d}\rangle \in I^2(a[d/x], A)\}$;

(d) $I^2(a, \bot) = \varnothing$;

(e) $I^2(a, \varphi \to \psi) = D_s - (I^2(a,\varphi) - I^2(a,\psi))$;

(f) $I^2(a, \forall x_\alpha \varphi) = \bigcap_{d \in D_\alpha} I^2(a[d/x], \varphi)$;

(g) $I^2(a, \langle R\rangle\varphi) = \{w \mid \exists w' \text{ with } \langle w, w'\rangle \in I^2(a,R) \text{ and } w' \in I^2(a,\varphi)\}$;

(h) $I^2(a, R^{\smile}) = \{\langle w, w'\rangle \mid \langle w', w\rangle \in I^2(a,R)\}$.

The definition obviously follows definition 17 save in its first clause. Note that well-definedness does not depend on the question whether I is an intension function for D and $\mathcal{L}$, and indeed the latter is not immediately obvious. We need to check the requirements in definition 13. That $I(a,x) = a(x)$ for any variable x is immediate and that $I(a,A) = I(a',A)$ if a and a' agree on the variables free in A follows by a standard property of substitutions and an easy induction. Suppose that B is free for x in A. Then

$$I^1(a, A\{x := B\}) = A\{x := B\}^{\overleftarrow{a}} = A^{\overleftarrow{a}}[x := B^{\overleftarrow{a}}] =$$
$$A^{\overleftarrow{a}}[x := I^1(a,B)] = A^{\overleftarrow{a[I(a,B)/x]}} = I^1(a[I(a,B)/x], A)\ .$$

That $I^2(a, A\{x := B\}) = I^2(a[I(a,B)/x], A)$ follows by a straightforward induction on the complexity of A which we leave to the reader. It follows that $I(a, A\{x := B\}) = I(a[I(a,B)/x], A)$ if B is free for x in A.

It remains to be shown that $I(a,A) \in D_\alpha$ for any assignment a and term A of type α. This is done by induction on the complexity of A. That $I(a, x_\alpha) \in D_\alpha$ if x is a variable follows from the stipulation that $I(a,x) = a(x)$ and that $I(a, c_\alpha) \in D_\alpha$ if α is basic is immediate. In the remaining cases the type of A is complex. Since it is easily seen by a separate induction that $I^2(a, A_\alpha) \subseteq D_s \times D_{\alpha_1} \times \cdots \times D_{\alpha_n}$ if $\alpha = \langle \alpha_1 \ldots \alpha_n\rangle$, it suffices to prove that, whenever $\alpha = \langle \alpha_1 \ldots \alpha_n\rangle$, and $\langle B_1, e_1\rangle \in D_{\alpha_1}, \ldots, \langle B_n, e_n\rangle \in D_{\alpha_n}$

(a) If $\mathsf{T}u\colon A^{\overleftarrow{a}} B_1 \ldots B_n \in \Gamma$ then $\langle\langle u,u\rangle, \langle B_1, e_1\rangle, \ldots, \langle B_n, e_n\rangle\rangle \in I^2(a,A)$;

(b) If $\mathsf{F}u\colon A^{\overleftarrow{a}} B_1 \ldots B_n \in \Gamma$ then $\langle\langle u,u\rangle, \langle B_1, e_1\rangle, \ldots, \langle B_n, e_n\rangle\rangle \notin I^2(a,A)$.

We will consider some remaining cases of the induction, leaving others to the reader. IH will be short for 'induction hypothesis'.

- $A_\alpha \equiv c$ and $\alpha = \langle \alpha_1 \ldots \alpha_n\rangle$. The requirement follows from the definition of $I^2(a,c)$ and clause (i) of definition 27.

- $A \equiv B_{\langle \beta\alpha_1 \ldots \alpha_n\rangle} C_\beta$. Suppose $\langle B_1, e_1\rangle \in D_{\alpha_1}, \ldots, \langle B_n, e_n\rangle \in D_{\alpha_n}$. From the induction hypothesis it follows that $I(a,C) = \langle C^{\overleftarrow{a}}, I^2(a,C)\rangle \in D_\beta$. Hence

$$\begin{array}{rcl}
\mathsf{T}u\colon (BC)^{\overleftarrow{a}} B_1 \ldots B_n \in \Gamma & \iff & \\
\mathsf{T}u\colon B^{\overleftarrow{a}} C^{\overleftarrow{a}} B_1 \ldots B_n \in \Gamma & \Longrightarrow & \text{(IH)} \\
\langle\langle u,u\rangle, I(a,C), \langle B_1, e_1\rangle, \ldots, \langle B_n, e_n\rangle\rangle \in I^2(a,B) & \iff & \text{(def. of } I\text{)} \\
\langle\langle u,u\rangle, \langle B_1, e_1\rangle, \ldots, \langle B_n, e_n\rangle\rangle \in I^2(a,BC) & &
\end{array}$$

This proves the (a) part of the case; the (b) part is similar.

- $A \equiv (\lambda x_{\alpha_1} C_{\langle \alpha_2 \dots \alpha_n \rangle})$. Again suppose $d_i = \langle B_i, e_i \rangle \in D_{\alpha_i}$, and reason as follows.

$$
\begin{array}{rcl}
\mathsf{F}u\colon (\lambda x.C)\overleftarrow{a}\, B_1 \dots B_n \in \Gamma & \iff & \\
\mathsf{F}u\colon (\lambda x.C\overleftarrow{a}\,[x := x])B_1 \dots B_n \in \Gamma & \Longrightarrow & \text{def. 27, } B_1 \text{ is closed} \\
\mathsf{F}u\colon C\overleftarrow{a}\,[x := B_1]B_2 \dots B_n \in \Gamma & \iff & \\
\mathsf{F}u\colon C\overleftarrow{a[d_1/x]}B_2 \dots B_n \in \Gamma & \Longrightarrow & \text{(IH)} \\
\langle \langle u, u \rangle, d_2, \dots, d_n \rangle \notin I^2(a[d_1/x], C) & \iff & \text{(def. of } I^2) \\
\langle \langle u, u \rangle, d_1, d_2, \dots, d_n \rangle \notin I^2(a, \lambda x.C) & &
\end{array}
$$

  This proves the (b) part, which is similar to the (a) part.

- $A_{\langle\rangle} \equiv \forall x_\beta \varphi$. Let $d \in D_\beta$ be arbitrary. Then $d = \langle B, e \rangle$ for some closed term B. In order to prove the (a) part of the statement we reason as follows.

$$
\begin{array}{rcl}
\mathsf{T}u\colon (\forall x \varphi)\overleftarrow{a} \in \Gamma & \iff & \\
\mathsf{T}u\colon \forall x \varphi\overleftarrow{a}\,[x := x] \in \Gamma & \Longrightarrow & \text{def. 27} \\
\mathsf{T}u\colon \varphi\overleftarrow{a}\,[x := x]\{x := B\} \in \Gamma & \iff & \\
\mathsf{T}u\colon \varphi\overleftarrow{a[d/x]} \in \Gamma & \Longrightarrow & \text{(IH)} \\
\langle u, u \rangle \in I^2(a[d/x], \varphi) & &
\end{array}
$$

  Since d was arbitrary, we conclude that $\langle u, u \rangle \in I^2(a, \forall x \varphi)$. The (b) part is similar.

- The cases $A_{\langle\rangle} \equiv \bot$, $A_{\langle\rangle} \equiv \varphi \to \psi$, $A_{\langle\rangle} \equiv \langle R \rangle \varphi$ and $A_{\langle s \rangle} \equiv R^{\smile}$ are straightforward.

This concludes the proof that I is an intension function for D and $\mathcal{L}$. Now define the function E by letting $E(\langle A, e \rangle) = e$ if $\langle A, e \rangle \in D_\alpha$ for any complex α. Clearly, $E(I(a, A)) = I^2(a, A)$ for any A, E is an extension function for D, and $M = \langle D, I, E \rangle$ is an intensional model for the language $\mathcal{L}$. It is easy to see that M satisfies Γ.

In order to prove the second part of the lemma, assume that Γ is complete. We have already established that M is countable in that case, and proposition 26 gives a normal countable intensional model satisfying Γ. ❑

We now come to the first step sketched in the introduction to this section. The notion of an *unsatisfiability property* (related to the *abstract consistency properties* of Smullyan [48] and Fitting [17]) is defined and it is shown that branches that do not have a 'sound' unsatisfiability property can in fact be extended to a downward saturated branch and hence are satisfiable. The interest in the theorem comes from the fact that many interesting notions can in fact be related to sound unsatisfiability properties as we shall see below.

DEFINITION 29. Let $\mathcal{U}$ be a set of branches in the language $\mathcal{L}$. $\mathcal{U}$ is an *unsatisfiability property* in $\mathcal{L}$ if, for each tableau rule $\Gamma/\Gamma_1; \dots; \Gamma_n$, $\{\Gamma_1, \dots, \Gamma_n\} \subseteq \mathcal{U}$ implies $\Gamma \in \mathcal{U}$.

An unsatisfiability property $\mathcal{U}$ in $\mathcal{L}$ is *sound* if no $\Gamma \in \mathcal{U}$ is satisfied by an intensional model for $\mathcal{L}$.

THEOREM 30 (Model Existence). *Let $\mathcal{L}$ and $\mathcal{C}$ be languages such that each $\mathcal{C}_\alpha$ is denumerably infinite and $\mathcal{L} \cap \mathcal{C} = \varnothing$. Assume that $\mathcal{U}$ is a sound unsatisfiability property in $\mathcal{L} \cup \mathcal{C}$ and that Γ is a branch in the language $\mathcal{L}$. If $\Gamma \notin \mathcal{U}$ then Γ is satisfied by a countable normal intensional model.*

Proof. Let $\mathcal{U}$ and Γ be as described. We construct a downward saturated branch Γ^* such that $\Gamma \subseteq \Gamma^*$. Let $\vartheta_1, \ldots, \vartheta_n, \ldots$ be an enumeration of all labeled sentences in $\mathcal{L} \cup \mathcal{C}$. Write $\#(\vartheta)$ for the index that the labeled sentence ϑ obtains in this enumeration. Let $\Gamma_0 = \Gamma$ and define each Γ_{n+1} by distinguishing the following four cases.

- $\Gamma_{n+1} = \Gamma_n$, if $\Gamma_n \cup \{\vartheta_n\} \in \mathcal{U}$;
- $\Gamma_{n+1} = \Gamma_n \cup \{\vartheta_n\}$, if $\Gamma_n \cup \{\vartheta_n\} \notin \mathcal{U}$ and ϑ_n is not of the form $\mathsf{F}u\colon \forall x\varphi$ or of the form $\mathsf{T}u\colon \langle R\rangle\varphi$;
- $\Gamma_{n+1} = \Gamma_n \cup \{\vartheta_n, \mathsf{F}u\colon \varphi\{x := c\}\}$, where c is the first constant in $\mathcal{C}_\alpha$ which does not occur in $\Gamma_n \cup \{\vartheta_n\}$, if $\Gamma_n \cup \{\vartheta_n\} \notin \mathcal{U}$ and $\vartheta_n \equiv \mathsf{F}u\colon \forall x_\alpha\varphi$;
- $\Gamma_{n+1} = \Gamma_n \cup \{\vartheta_n, \mathsf{T}u\colon Ru', \mathsf{T}u'\colon \varphi\}$, where u' is the first constant in $\mathcal{C}_s$ which does not occur in $\Gamma_n \cup \{\vartheta_n\}$, if $\Gamma_n \cup \{\vartheta_n\} \notin \mathcal{U}$ and $\vartheta_n \equiv \mathsf{T}u\colon \langle R\rangle\varphi$.

This is well-defined since each Γ_n contains only a finite number of constants from $\mathcal{C}$. That $\Gamma_n \notin \mathcal{U}$ for each n follows by a simple induction which uses the definition of an unsatisfiability property and the fact that $\mathsf{F}\forall$ and $\mathsf{T}\langle\cdot\rangle$ are tableau rules. Define $\Gamma^* = \bigcup_n \Gamma_n$. For all finite sets $\{\vartheta_{k_1}, \ldots, \vartheta_{k_n}\}$ and all $k \geq \max\{k_1, \ldots, k_n\}$

(20) $\{\vartheta_{k_1}, \ldots, \vartheta_{k_n}\} \subseteq \Gamma^* \Leftrightarrow \Gamma_k \cup \{\vartheta_{k_1}, \ldots, \vartheta_{k_n}\} \notin \mathcal{U}$

In order to show this, let $k \geq \max\{k_1, \ldots, k_n\}$ and let $\{\vartheta_{k_1}, \ldots, \vartheta_{k_n}\} \subseteq \Gamma^*$. Then there is some ℓ such that $\{\vartheta_{k_1}, \ldots, \vartheta_{k_n}\} \subseteq \Gamma_\ell$. Let $m = \max\{k, \ell\}$. We have that $\Gamma_k \cup \{\vartheta_{k_1}, \ldots, \vartheta_{k_n}\} \subseteq \Gamma_m$. Since $\Gamma_m \notin \mathcal{U}$ and $\mathcal{U}$ is closed under supersets (because of the weakening rule W), it follows that $\Gamma_k \cup \{\vartheta_{k_1}, \ldots, \vartheta_{k_n}\} \notin \mathcal{U}$. For the reverse direction, suppose that $\Gamma_k \cup \{\vartheta_{k_1}, \ldots, \vartheta_{k_n}\} \notin \mathcal{U}$. Then, since $\mathcal{U}$ is closed under supersets, $\Gamma_{k_i} \cup \{\vartheta_{k_i}\} \notin \mathcal{U}$, for each of the k_i. By the construction of Γ^* each $\vartheta_{k_i} \in \Gamma^*$ and $\{\vartheta_{k_1}, \ldots, \vartheta_{k_n}\} \subseteq \Gamma^*$.

Γ^* is a downward saturated branch. The conditions (g) and (h) of definition 27 immediately follow from the construction of Γ^*. For checking the other conditions of definition 27 the equivalence in (20) can be used. Here we check condition (i), which may serve as an example for the other cases. Assume $\mathsf{F}u\colon \langle R\rangle\varphi \in \Gamma^*$ and let u' be a constant of type s. Let k be the maximum of $\#(\mathsf{F}u\colon \langle R\rangle\varphi)$, $\#(\mathsf{F}u\colon Ru')$, and $\#(\mathsf{F}u'\colon \varphi)$. Since, by (20), $\Gamma_k \cup \{\mathsf{F}u\colon \langle R\rangle\varphi\} \notin \mathcal{U}$, it must be the case by definition 29 and the fact that $\mathsf{F}\langle\cdot\rangle$ is a tableau rule that either $\Gamma_k \cup \{\mathsf{F}u\colon Ru'\} \notin \mathcal{U}$ or $\Gamma_k \cup \{\mathsf{F}u'\colon \varphi\} \notin \mathcal{U}$. Using (20) again, we find that either $\mathsf{F}u\colon Ru' \in \Gamma^*$ or $\mathsf{F}u'\colon \varphi \in \Gamma^*$.

We conclude that Γ^* is satisfied by an intensional model M. In order to prove that there is a normal countable intensional model satisfying Γ^* and hence Γ it suffices to show that Γ^* is complete. Let φ be any sentence of $\mathcal{L} \cup \mathcal{C}$ and assume that $\mathsf{T}u\colon \varphi \notin \Gamma^*$ and $\mathsf{F}u\colon \varphi \notin \Gamma^*$. Then, by (20), $\Gamma_k \cup \{\mathsf{T}u\colon \varphi\} \in \mathcal{U}$ and $\Gamma_k \cup \{\mathsf{F}u\colon \varphi\} \in \mathcal{U}$, for sufficiently large k. But M satisfies Γ_k and therefore must either satisfy $\Gamma_k \cup \{\mathsf{T}u\colon \varphi\}$ or $\Gamma_k \cup \{\mathsf{F}u\colon \varphi\}$, contradicting the soundness of $\mathcal{U}$. Thus Γ^* is complete and some normal countable intensional model satisfies Γ^* and Γ. ❑

We can now reap a harvest of corollaries by relating various notions to unsatisfiability properties. In the following Γ will always be a branch in some language $\mathcal{L}$ while Δ ranges over branches in $\mathcal{L} \cup \mathcal{C}$, where $\mathcal{L}$ and $\mathcal{C}$ are as in the formulation of Theorem 30.

COROLLARY 31 (Generalized Compactness). *If each finite* $\Gamma_0 \subseteq \Gamma$ *is satisfiable then* Γ *is satisfiable.*

Proof. $\{\Delta \mid \text{some finite } \Delta_0 \subseteq \Delta \text{ is unsatisfiable}\}$ is a sound unsatisfiability property. ❑

COROLLARY 32 (Generalized Löwenheim–Skolem). *If* Γ *is satisfiable then* Γ *is satisfiable by a countable intensional model.*

Proof. $\{\Delta \mid \Delta \text{ is unsatisfiable}\}$ is a sound unsatisfiability property. ❑

COROLLARY 33 (Generalized Completeness). *If* Γ *is unsatisfiable then* Γ *has a closed tableau.*

Proof. $\{\Delta \mid \Delta \text{ has a closed tableau}\}$ is a sound unsatisfiability property. ❑

COROLLARY 34 (Admissability of Cut). *If* $\Gamma \cup \{\mathsf{T}u{:}\, \varphi\}$ *and* $\Gamma \cup \{\mathsf{F}u{:}\, \varphi\}$ *have closed tableaus then* Γ *has a closed tableau.*

Proof. Use soundness and generalized completeness. ❑

7 CONCLUSION

This chapter has looked at some of the motivations for combining modality with quantification and abstraction over objects of higher order. Montague's logic **IL** was reviewed and was found to have some shortcomings: it is not Church-Rosser and it is not intensional in Whitehead and Russell's original sense. An alternative higher order modal logic **MTT** was then introduced. **MTT** imports many ideas from the higher order logics in Fitting [19], but is based on a simpler notion of model. We have dubbed the generalized models on which **MTT** is based *intensional* models. As was shown above, the usual rules of α, β and η conversion can consistently be added to the logic in which case the logic sports the Church-Rosser property.

The logic is also fully intensional (or "hyperintensional") in the sense that co-entailing expressions need not be identical and we shall use the rest of this conclusion to discuss some points that arise in relation with this. Consider (21–24), where in each case a natural language sentence is accompanied by its **MTT** rendering. (Here *fido*, *fritz* and *mary* are constants of individual type e, *in* is a predicate of type $\langle e \rangle$, and *know* is a relation of type $\langle\langle\rangle e\rangle$.)

(21) a. Fritz is out if Fido is in
b. $\mathit{in}\,\mathit{fido} \rightarrow \neg(\mathit{in}\,\mathit{fritz})$

(22) a. Fido is out if Fritz is in
b. $\mathit{in}\,\mathit{fritz} \rightarrow \neg(\mathit{in}\,\mathit{fido})$

(23) a. Mary knows Fritz is out if Fido is in
b. $\mathit{know}\,(\mathit{in}\,\mathit{fido} \rightarrow \neg(\mathit{in}\,\mathit{fritz}))\;\mathit{mary}$

(24) a. Mary knows Fido is out if Fritz is in
b. $\mathit{know}\,(\mathit{in}\,\mathit{fritz} \rightarrow \neg(\mathit{in}\,\mathit{fido}))\;\mathit{mary}$

Simple tableaus will verify that (21b) and (22b) co-entail, as they should. But (23b) and (24b) do *not* co-entail: Note that $\{\mathsf{T}u\colon(23\text{b}), \mathsf{F}u\colon(24\text{b})\}$ is downward saturated and thus will have an intensional model refuting one direction of the entailment.

It may be protested that there is at least one sense in which Mary knows that Fido is out if Fritz is in if she knows that Fritz is out if Fido is in: While she may have failed to derive the contraposed statement *explicitly*, there is still a sense in which she is *implicitly* committed to it. Such a notion of *implicit knowledge* is also available in **MTT**. Let K be a constant of type $\langle es\rangle$. K can be given the role of an *epistemic alternative relation* by adopting the following meaning postulate.

(25) $\Box\forall x_e\forall w_s(Kxw \leftrightarrow \forall p_{\langle\rangle}(\mathit{know}\,px \rightarrow \Diamond(\dot{w}\wedge p)))$

This says effectively that a world w is an epistemic alternative for a person x if w is in the intersection of the extensions of all propositions that x explicitly knows to hold.[8] A tableau will show that (25) entails (26).

(26) $\forall x_e\forall p_{\langle\rangle}(\mathit{know}\,px \rightarrow [Kx]p)$

Thus it can be deduced that (27), where the modal operator $[K\,\mathit{mary}]$ was used to model Mary's implicit beliefs, follows from (23). In fact *implicit* beliefs are closed under consequence and hence co-entailment.

(27) a. Mary implicitly knows Fido is out if Fritz is in
b. $[K\,\mathit{mary}](\mathit{in}\,\mathit{fritz} \rightarrow \neg(\mathit{in}\,\mathit{fido}))$

The non-equivalence of (23b) and (24b) discussed above illustrates that **MTT** is intensional in Whitehead and Russell's sense of the term. Relations, including zero-place relations, can be co-extensional without being identical. This means that linguistic expressions that are assumed to denote relations are no longer predicted to be intersubstitutable if they have the same extension, not even if they have the same extension in all possible worlds.

This is not unimportant since many expressions in natural language are undoubtedly relational and a nasty foundational problem will no longer be associated with them, but there seems to be a rest category of problems with expressions of *basic* type. Above we have treated *proper names* as having a basic type e, and this leads to the well-known Hesperus–Phosphorus, or Cicero–Tully, kind of problem. If *Hesperus* is translated as $\mathit{hesperus}_e$, *Phosphorus* as $\mathit{phosphorus}_e$, and the identity statement *Hesperus is Phosphorus* as $\mathit{hesperus} = \mathit{phosphorus}$, the consequence will be the false prediction that the two names can be substituted for one another in any context *salva veritate*.

There are two reactions to this. One possible reaction is an adaptation of the logic. One could introduce some domain of individual concepts and allow many-one correspondences between individual concepts and individuals. Such a strategy is followed by Fox and Lappin [20] in a different set-up, but in our case it would lead to a complication of the logic, be it probably a mild one.

The second reaction leaves the logic as is, but adapts the rendering of natural language expressions. If names can be treated as predicates in some way, the intension–extension

[8] Note that the present set-up distinguishes between propositions (the elements of $D_{\langle\rangle}$) and sets of possible worlds. The extension of a proposition will be a set of worlds. Different propositions may determine the same extension.

distinction will come for free for them as well. In fact, the existing literature contains several proposals for treating names as based on predicates and not on individual constants. Russell's description theory of names is an early example and Montague [38] offers another example by essentially treating names as being of the "raised" type $\langle\langle e\rangle\rangle$, not simply of type e. In combination with a treatment of identity as co-extensionality (in all possible worlds) this would avoid the problems if our logic is used. A third proposal that in effect treats names as relations comes from the literature on plurality. Many authors on this subject, starting with Bartsch [5] and Bennett [6] (see Lønning [34] for an overview), have argued that both singular and plural individuals should in fact be treated as sets, with the semantic property of being a singleton corresponding to the grammatical notion of singularity. In the present set-up this effect can be obtained by redefining type e as a complex type $\langle 0\rangle$, where 0 is a new basic type for abstract individuals. Type 0 objects will now correspond one-to-one with the *extensions* of those type e objects that have singleton extensions, i.e. to singular individuals, but there are many intensional models in which $\mathit{hesperus}_e$ and $\mathit{phosphorus}_e$ are co-extensional (with a singleton extension) in all worlds but are not identical. Let $A_e \approx B_e$ be an abbreviation of $\Box\forall x_0(Ax \leftrightarrow Bx)$, i.e. let $A \approx B$ express necessary co-extensionality, and assume that natural language *is* (the "*is* of identity") in fact expresses $\approx$. Then the argument in (28) will be rendered as (29) and will therefore be predicted to be invalid.

$$\text{(28)}\quad \frac{\text{Hesperus is Phosphorus} \qquad \text{Mary knows that Hesperus is Hesperus}}{\text{Mary knows that Hesperus is Phosphorus}}$$

$$\text{(29)}\quad \frac{\mathit{hesperus} \approx \mathit{phosphorus} \qquad \mathit{know}\,(\mathit{hesperus} \approx \mathit{hesperus})\ \mathit{mary}}{\mathit{know}\,(\mathit{hesperus} \approx \mathit{phosphorus})\ \mathit{mary}}$$

Again, the invalidity of the argument depends on the fact that Mary's knowledge was taken to be Mary's *explicit* knowledge. If implicit knowledge is taken, the argument will turn out to be valid, as the reader will have no difficulty to verify.

We conclude that the logic **MTT** is truly intensional, as it will distinguish between the meaning of one relation and another necessarily co-extensive with it. This can be used to avoid many substitution problems in natural language semantics and other areas. If it is moreover accepted that proper names should in fact be treated as constants of complex type, they will also be treated hyperintensionally. For example, letting them be of type $\langle 0\rangle$, a move which may be argued for on independent grounds having to do with the treatment of plurality, will make them start to act as naming individual concepts and substitution problems with them will be avoided.

ACKNOWLEDGEMENTS

I would like to thank Grigori Mints for his valuable comments. The editors I thank for their comments, their patience, and their encouragement.

BIBLIOGRAPHY

[1] P. B. Andrews. Resolution in Type Theory. *Journal of Symbolic Logic*, 36(3):414–432, 1971.
[2] P. B. Andrews. General Models and Extensionality. *Journal of Symbolic Logic*, 37(2):395–397, 1972.

[3] P. B. Andrews. *An Introduction to Mathematical Logic and Type Theory: to Truth through Proof.* Academic Press, Orlando, Florida, 1986.

[4] H. Barendrecht. *The Lambda Calculus: its Syntax and Semantics.* North-Holland, 1981.

[5] R. Bartsch. The Semantics and Syntax of Number and Numbers. In J.P. Kimball, editor, *Syntax and Semantics vol. 2*, pages 51–93. Seminar Press, New York, 1973.

[6] M. Bennett. *Some Extensions of a Montague Fragment of English.* PhD thesis, UCLA, 1974. distributed by the Indiana University Linguistics Club.

[7] J.F.A.K. van Benthem. *Essays in Logical Semantics.* Reidel, Dordrecht, 1986.

[8] J.F.A.K. van Benthem and K. Doets. Higher-Order Logic. In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, Vol. I*, pages 275–329. Reidel, Dordrecht, 1983.

[9] C. Benzmüller, C. E. Brown, and M. Kohlhase. Higher Order Semantics and Extensionality. *Journal of Symbolic Logic*, 69, 2004.

[10] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press, Cambridge, 2001.

[11] A. Bressan. *A General Interpreted Modal Calculus.* Yale University Press, New Haven and London, 1972.

[12] R.A. Bull. On Modal Logic with Propositional Quantifiers. *Journal of Symbolic Logic*, 34:257–263, 1969.

[13] B. ten Cate. Expressivity of Second-order Modal Logic. *Journal of Philosophical Logic*, to appear.

[14] A. Church. A Formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[15] D.R. Dowty, R.E. Wall, and S. Peters. *Introduction to Montague Semantics.* Reidel, Dordrecht, 1981.

[16] K. Fine. Propositional Quantifiers in Modal Logic. *Theoria*, 36:336–346, 1970.

[17] M. Fitting. *First-Order Logic and Automated Theorem Proving.* Springer, New York, 1996.

[18] M. Fitting. Interpolation for First Order **S5**. *Journal of Symbolic Logic*, 67(2):621–634, 2002.

[19] M. Fitting. *Types, Tableaus, and Gödels God.* Kluwer Academic Publishers, Dordrecht, 2002.

[20] C. Fox and S. Lappin. *Foundations of Intensional Semantics.* Blackwell, Oxford, 2005.

[21] J. Friedman and D. Warren. Lambda Normal Forms in an Intensional Logic for English. *Studia Logica*, 39:311–324, 1980.

[22] D. Gallin. *Intensional and Higher-Order Modal Logic.* North-Holland, Amsterdam, 1975.

[23] L.T.F. Gamut. *Logic, Language and Meaning.* University of Chicago Press, Chicago, 1991.

[24] K. Gödel. Ontological Proof. In S. Feferman, J.W. Dawson, W. Goldfarb, C. Parsons, and R.N. Solovay, editors, *Kurt Gödel Collected Works, Volume III, Unpublished Essays and Lectures*, pages 403–404. Oxford University Press, New York, 1995.

[25] Ph. de Groote. Towards Abstract Categorial Grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, pages 148–155, Toulouse, France, 2001. ACL.

[26] H.L.W.H. Hendriks. Type Change in Semantics: the Scope of Quantification and Coordination. In E. Klein and J.F.A.K. van Benthem, editors, *Categories, Polymorphism, and Unification*, pages 95–119. Centre for Cognitive Science, Edinburgh, 1988.

[27] H.L.W.H. Hendriks. *Studied Flexibility: Categories and Types in Syntax and Semantics.* PhD thesis, University of Amsterdam, 1993.

[28] L. Henkin. Completeness in the Theory of Types. *Journal of Symbolic Logic*, 15:81–91, 1950.

[29] L. Henkin. A Theory of Propositional Types. *Fundamenta Mathematicae*, 52:323–344, 1963.

[30] D. Kaplan. **S5** with Quantifiable Propositional Variables. *Journal of Symbolic Logic*, 35:355, 1970.

[31] Ph. Kremer. Defining Relevant Implication in a Propositionally Quantified **S4**. *Journal of Symbolic Logic*, 62(4):1057–1069, 1997.

[32] Ph. Kremer. On the Complexity of Propositional Quantification in Intuitionistic Logic. *Journal of Symbolic Logic*, 62(2):529–544, 1997.

[33] S. Kripke. A Completenes Theorem in Modal Logic. *Journal of Symbolic Logic*, 24:1–14, 1959.

[34] J.T. Lønning. Plurals and Collectivity. In J.F.A.K. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 1009–1053. Elsevier, Amsterdam, 1997.

[35] R. Montague. On the Nature of Certain Philosophical Entities. *Monist*, 53:161–194, 1969. Reprinted in [51].

[36] R. Montague. English as a Formal Language. In B. Visenti et al., editor, *Linguaggi nella Società e nella Tecnica*, pages 189–224. Edizioni di Comunità, Milan, 1970. Reprinted in [51].

[37] R. Montague. Universal Grammar. *Theoria*, 36:373–398, 1970. Reprinted in [51].

[38] R. Montague. The Proper Treatment of Quantification in Ordinary English. In J. Hintikka, J. Moravcsik, and P. Suppes, editors, *Approaches to Natural Language*, pages 221–242. Reidel, Dordrecht, 1973. Reprinted in [51].

[39] R.A. Muskens. *Meaning and Partiality.* CSLI, Stanford, 1995.

[40] R.A. Muskens. Categorial Grammar and Lexical-Functional Grammar. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG01 Conference, University of Hong Kong*, pages 259–279, Stanford CA, 2001. CSLI Publications.

[41] R.A. Muskens. Language, Lambdas, and Logic. In Geert-Jan Kruijff and Richard Oehrle, editors, *Resource Sensitivity in Binding and Anaphora*, Studies in Linguistics and Philosophy, pages 23–54. Kluwer, 2003.

[42] S. Orey. Model Theory for the Higher Order Predicate Calculus. *Transactions of the American Mathematical Society*, 92:72–84, 1959.

[43] B. Partee. Montague Grammar. In J.F.A.K. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 5–91. Elsevier, Amsterdam, 1997. with H. Hendriks.

[44] B. Partee and M. Rooth. Generalized Conjunction and Type Ambiguity. In R. Baüerle, C. Schwarze, and A. von Stechow, editors, *Meaning, Use and Interpretation of Language*, pages 361–383. de Gruyter, Berlin, 1983.

[45] D. Prawitz. Hauptsatz for Higher Order Logic. *Journal of Symbolic Logic*, 33(3):452–457, 1968.

[46] B. Russell. Mathematical Logic as Based on the Theory of Types. *American Journal of Mathematics*, 30:222–262, 1908.

[47] K. Schütte. Syntactical and Semantical Properties of Simple Type Theory. *Journal of Symbolic Logic*, 25(4):305–326, 1960.

[48] R.M. Smullyan. *First-Order Logic*. Springer-Verlag, Berlin, 1968.

[49] M. Takahashi. A Proof of Cut-elimination Theorem in Simple Type Theory. *Journal of the Mathematical Society of Japan*, 19(4):399–410, 1967.

[50] R. Thomason. A semantic analysis of tense logics. *Journal of Symbolic Logic*, 37(1):150–158, 1972.

[51] R. Thomason, ed. *Formal Philosophy, Selected Papers of Richard Montague*, Yale University Press, 1974.

[52] A. N. Whitehead and B. Russell. *Principia Mathematica*. Cambridge University Press, 1910-13.

[53] E. Zalta. *Abstract Objects: An Introduction to Axiomatic Metaphysics*. Reidel, Dordrecht, 1983.

[54] E. Zalta. *Intensional Logic and the Metaphysics of Intentionality*. MIT Press/Bradford Books, Cambridge, MA, 1988.

[55] E. Zalta. A Comparison of Two Intensional Logics. *Linguistics and Philosophy*, 11:59–89, 1988.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

11

TEMPORAL LOGIC

Ian Hodkinson and Mark Reynolds

1	Introduction	656
2	Structures	658
2.1	Structures for propositional temporal logic	658
2.2	Cycles	661
2.3	Branching Time	662
2.4	Granularity	665
2.5	Many-dimensional Evaluation	666
2.6	Intervals	666
2.7	Combining temporal logic with other logics	667
3	Temporal logical systems	668
3.1	First-order logic	669
3.2	Monadic second-order logic	669
3.3	Temporalised first-order logic	670
3.4	Temporal Logics	671
3.5	Extensions of temporal logic	674
3.6	Temporal logic with second-order operations	677
3.7	Branching Time Operations	680
4	Expressive power of internal and external paradigms	682
4.1	Translating temporal logic into first-order logic	683
4.2	Expressive completeness	683
4.3	The class of all flows of time	685
4.4	Linear flows — Kamp's theorem	685
4.5	Computational issues	686
4.6	Separation	687
4.7	Many-dimensional temporal logic	690
4.8	First-order temporal logic	691
5	Temporal reasoning	696
5.1	Hilbert style axiom systems	696
5.2	Gentzen systems	702
5.3	Natural Deduction	702
5.4	Tableaux	702
5.5	Resolution	704
5.6	Automata	704
5.7	Translation into first-order logic	705
5.8	Filtration and the finite model property	706
5.9	Other modal methods	708
5.10	Mosaics	708
5.11	Monodic fragments of first-order temporal logic	710
6	Conclusion	712

1 INTRODUCTION

Time has always been with us, though few of us have enough of it. The nature of time itself is a conundrum that we nowadays leave to physicists. But we have always had to find our way through time, plan our activities, and cope with the uncertain future. This can be, indeed, has to be done without a deep scientific knowledge of what makes time tick.

We use language and its rich tense structure to express and reason about events in time. This of course throws up linguistic and philosophical conundrums of its own. With the rise in the 20th century of formal logical languages, it became natural to try to express temporal concepts and arguments in formal terms, and so it was that Arthur Prior from the 1950s came to develop *tense logics.* These were modal logics, with box-modalities H and G for 'always in the past' and 'always in the future', motivated by tenses in natural language. The advent of Kripke semantics in the 1960s gave the enterprise a boost, because a Kripke frame is so naturally seen as a set of time points endowed with an 'earlier-later' relation.

Temporal logic today is a large, busy subject with stakeholders from many disciplines. Philosophers and linguists have continued to make major contributions to it. Since Pnueli's pioneering 1977 paper [147], several branches of computer science and related fields — such as databases, specification and verification, synthesis of programs, temporal planning, temporal knowledge representation — have had a huge influence, and the use of temporal logic in some of these areas has developed to the point of commercial application. There is even some contact with physics, but so far this has been limited.

Temporal logic is in a way a branch of applied modal logic, but modal logicians may be disconcerted by what they find here. Temporal logic has always focused on handling time, it has developed whatever methods it found useful for this end, and not all of them are modal in a narrow sense. Connectives such as Until and Since, again mimicking the natural language constructs, go beyond boxes and diamonds and are of great importance in the subject. Indeed, completely general first-order-definable connectives are used as well. Bearing in mind the evaluation and reference points of natural language, it is natural that many-dimensional evaluation has long been of importance in temporal logic, whereas it only recently attracted great interest in modal logic proper. The focus in temporal logic is on a fairly narrow range of Kripke frames — nearly always irreflexive and transitive, and typically linear orders or trees, though relativistic and circular time are sometimes considered. The natural numbers are the dominant model of linear time, though dense and continuous and indeed arbitrary linear orders have found their way in (and in this chapter we are happy to consider them). Sometimes the pressures of time have led to a style of evaluation of formulas that seems non-modal at first sight (see Section 3.7). A very influential strand of work, started by Kamp in 1968, compares the expressive power of modal and first-order languages on the model (rather than frame) level. Rather than be content with limited but well-behaved modal expressiveness, the thrust of the work created temporal languages as strong as classical first-order logic and even monadic second-order logic. Perhaps because the proofs rely heavily on the assumption that time is linear or even natural number-like, not much similar research in modal logic has been done. Classical logic is not just a benchmark for the expressiveness of 'real' temporal logics: using first-order logic for handling time is itself a respectable tradition. In temporal logic there is an unusual (for modal logic) use of methods from

classical mathematical logic and combinatorial techniques such as automata. Problems such as model-checking (considered in Chapter 17) are all-important in temporal logic but do not appear much in modal logic.

Nonetheless, from Prior's work onwards, modal ideas have been prominent in temporal logic. Its most basic syntax and Kripke semantics are (multi-)modal; one often comes across modal techniques such as canonicity and Sahlqvist's theorem, filtration and non-standard inference rules; and problems of axiomatisation, decidability, and complexity are ubiquitous in modal and temporal logic. Sophisticated results on modal logics above K4 have been transferred to temporal logic. Chapters 2, 3, 4, 9, and 12 are very relevant to temporal logic. In Chapter 17, the reader will find a concentrated discussion of modal and temporal logic in computer science. In the current chapter, we will examine some topics in temporal logic that are considered both in computer science and in other fields. As we have not the space to provide a rigorous development from scratch, the chapter is intended more as a gateway to the subject. It is mostly a survey-style commentary on some important strands, with directions to the literature for those wishing to find out more. Our priority is range rather than depth, but we cannot be comprehensive. A chapter of definitions would be indigestible, so we have tried to include some of the arguments, but space limitations have meant that their level of detail veers wildly from a few words to (occasionally) something approaching a full proof. Readers may of course skip details if they so desire.

We start out in Section 2 with a basic round-up of the semantic options for handling time. In Section 3 we cover some of the logics (syntax and evaluation) that can be used. Bearing in mind the remarks above, it will be no surprise that we do not confine ourselves to modal-style logics: first- and second-order logics, and others, find their way in, and our lack of consideration of mu-calculi is only because chapter 12 is devoted to them. Cn Section 4 we compare the expressivity of classical and modal-style logics. Kamp's famous 1968 expressive completeness theorem makes its appearance here. In Section 5 we discuss temporal reasoning, mainly avoiding automata (see Chapter 17 for them) but covering Hilbert systems, tableaux, resolution, filtration and the finite model property, and other methods.

A word about first-order temporal logic. This is a complex issue. There is a confusing variety of ways to add first-order logic to a temporal system, and undecidability results obtained in the 1960s, accompanied by later expressive incompleteness results, also cast their shade over the development of this part of the subject. But at the time of writing, there is something of a resurgence of interest in it from the database and reasoning communities. We will discuss the rudiments of first-order temporal logic in Sections refchapter11:sec2–3, and also some of the recent results on expressive completeness and decidability in Sections refchapter11:sec4–5. Chapter 9 is also relevant of course.

Temporal logic, then, is a branch of applied logic that brings to bear a gamut of powerful methods from many fields to study time and temporal phenomena. It is not wholly modal, but rests on a modal base — it is a meeting ground for concepts from modal logic, classical first-order logic, and higher-order logic. It has found very successful application in computing, and embodies seminal contributions from philosophy and linguistics as well. We hope our chapter, and other chapters here, will serve as a guide for the reader wishing to discover more about this intensely active field.

2 STRUCTURES

How can we model time? Clearly, such a question can generate much heated debate. Rather than make a futile attempt to settle it, we prefer to take a practical viewpoint, and simply offer the reader a number of options which have been studied in some depth and found useful.

This section is devoted to setting up some of the standard models of time. They will be called *structures.* We will use them as semantics for the various logics of time to be discussed in the next section. For now, we have no specific logic syntax in mind.

However, the future choice of syntax does have an effect here, because the structures we set up must be suitable for evaluating *atomic* formulas of the logics to come. So our treatment will divide into cases, according to the kind of evaluation we envisage. We will begin with the simplest case: models of time suitable for *propositional* temporal logics. This continues in Section 2.2 with cyclical models of time. In Section 2.3, we will consider some options for branching time. Section 2.4 will discuss structures supporting varying granularity of focus. Section 2.5 goes into the options when propositions depend on several time points; this leads naturally into Section 2.6, on temporal intervals. Finally, Section 2.7 considers the options for temporal logics beyond propositional.

2.1 Structures for propositional temporal logic

The simplest and most common form of temporal logic is *propositional temporal logic.* In it, time is viewed as simply a set of points. To facilitate making statements and reasoning about time, additional information is included in the model. We will start off with probably the simplest useful information, which is to state which time points are earlier than, or later than, which. To represent basic facts of interest, there are available a number of *atomic propositions,* or *propositional variables* (or as some say, *propositional atoms*). These are syntactic objects; they are usually written p, q, r, p_0, p_1, etc. Their truth values (true or false) are expected to be given by the model. These truth values will be time-dependent: so each atom will be either true or false at each time point, and the model will specify which. Logical machinery can be erected on top of the atoms in a variety of ways, to permit representation of and reasoning about more complex statements; this is the task of the next section.

Thus, our models or structures will have three parts: a set of time points; information about which time points come before or after which (this much is called a *flow of time*); and information about which atoms are true at which time points (this much is called a *temporal structure*).

Flows of time, and temporal structures

DEFINITION 1. A *flow of time* is a pair $(T, <)$, where T is a non-empty set, and $<$ is an irreflexive and transitive binary relation on T.

The idea is that T is the set of *time points,* and $<$ is the *earlier-later relation* on T. For time points $t, u \in T$, $t < u$ (or equally, $u > t$) will mean intuitively that t is earlier than u, and that u is later than t. This explains the requirements that $<$ be irreflexive — no time point should be in the past or future of itself — and transitive — if t is earlier than u, and u earlier than v, then we expect t to be earlier than v. However, we will

see a few circumstances (such as in Section 2.2) where it is appropriate to modify these requirements.

We can define the notion of two flows of time being isomorphic, or one being a subflow of the other, in the usual way. We also define the time point relations $>$, $\leq$, and $\geq$ in terms of $<$ in the usual way.

Clearly, a flow of time $(T, <)$ can be regarded as a (certain kind of) Kripke frame: the set of possible worlds is T, and the accessibility relation is $<$. This will later lead to modal-style logics for time.

Modelling truth and falsity of the atoms is done in the usual way. We assume a fixed ambient set L of atoms. Let $\wp(T)$ be the set of subsets of any given set T.

DEFINITION 2. A *temporal structure* is a triple $(T, <, h)$, where $(T, <)$ is a flow of time, and $h : L \to \wp(T)$ is a map (called an 'assignment' or 'valuation').

We regard an atom q as being true at a time $t \in T$ if $t \in h(q)$, and false at time t if $t \notin h(q)$. Some authors present valuations in the form $g : T \to \wp(L)$, $g(t)$ being the set of atoms regarded as true at time t. The two methods obviously carry the same information. Clearly, a temporal structure can be regarded as a Kripke structure (see Chapter 1).

Classes of flows of time We will often be interested in various *classes* of flows of time. The class of all flows of time is one such. Another is the class of all linear flows of time:

DEFINITION 3. A flow of time $(T, <)$ is said to be *linear* if given any two distinct time points in it, one is before the other. That is, $(T, <) \models \forall xy(x = y \vee x < y \vee y < x)$.

Linear flows have been very heavily studied, and various classes of linear flows will figure prominently in this chapter (though not to the exclusion of other kinds of flow). Here are some other interesting properties that a linear flow of time may have:

DEFINITION 4. Let $(T, <)$ be a linear flow of time.

1. $(T, <)$ is said to be *discrete* if for each $t \in T$,
   (a) if there is any $u \in T$ with $u > t$, then there is a first such u: one such that there is no $v \in T$ with $t < v < u$, and
   (b) if there is any $u \in T$ with $u < t$, then there is a last such u: one such that there is no $v \in T$ with $u < v < t$.
2. $(T, <)$ is said to be *dense* if for all $t, u \in T$, if $t < u$ then there is $v \in T$ with $t < v < u$. This is in some way the opposite of discrete.
3. $(T, <)$ is said to be *Dedekind complete* if any non-empty subset $S \subseteq T$ that is bounded above — i.e., there is $t \in T$ with $t \geq s$ for all $s \in S$ — has a least upper bound in T: i.e., there is $t \in T$ such that (i) $t \geq s$ for all $s \in S$, and (ii) there is no $t' < t$ with $t' \geq s$ for all $s \in S$. Equivalently, any non-empty $U \subseteq T$ that is bounded below has a greatest lower bound in T. (Any greatest lower bound of U is a least upper bound of $\{t \in T : t \leq u \text{ for all } u \in U\}$, and vice versa.)
4. $(T, <)$ is said to be *continuous* if it is dense and Dedekind complete.

5. $(T, <)$ is said to be *separable* if there is a countable subset $D \subseteq T$ that is *dense in* T*:* for all $t, u \in T$ with $t < u$, there is $v \in D$ with $t < v < u$. (It follows that $(T, <)$ is itself dense.)

The corresponding *classes* of linear flows, such as the dense linear ones, the discrete linear ones, and the continuous linear ones, as well as others such as the class of all finite linear flows, will be important to us. We will later write $\mathcal{L}$ for the class of all linear flows of time, and $\mathcal{D}$ for the class of all Dedekind-complete linear flows.

Non-linear flows such as trees are also much used:

DEFINITION 5. A flow of time $(T, <)$ is said to be a *tree* if for all $t \in T$, the set $\{u \in T : u < t\}$ is linearly ordered by $<$. A *branch* of a tree $(T, <)$ is a maximal linearly-ordered subset of T.

In a tree, the past of any time point is linear. However, its future may not be, so that many branches may pass through (i.e., contain) any given time point. Our models of 'branching time' will be based on trees.

There are also certain specific flows of time that are natural to consider, well studied, and useful in applications. We now list some of them, and make some comments. A general reference for information about linear orders is [171], in which the reader may find more details.

1. The natural numbers, $(\mathbb{N}, <)$, where $\mathbb{N} = \{0, 1, 2, \ldots\}$ and $<$ is the usual order. This is the most commonly used flow of all. It occurs naturally in computing applications, where programs execute instructions at successive moments. Time is viewed as discrete (ticking, so that any non-final moment has a next moment), linear, and with a first moment but no last moment. Of course, these properties alone are insufficient to pin down $\mathbb{N}$: for example, they are also true of the flow of time consisting of a copy of $\mathbb{N}$ followed by a copy of the integers, $\mathbb{Z}$, and indeed, this flow is indistinguishable from $(\mathbb{N}, <)$ by any first-order sentence. One may characterise $(\mathbb{N}, <)$ up to isomorphism in a second-order way as the unique discrete Dedekind-complete linear flow with a first point and no last point.

2. The integers, $(\mathbb{Z}, <)$, where $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$, and $<$ is the usual order. The discrete Dedekind-complete linear flows of time are precisely the ones that are isomorphic to sub-flows of $(\mathbb{Z}, <)$, and $(\mathbb{Z}, <)$ is up to isomorphism the unique one of these without endpoints.

3. The rationals, $(\mathbb{Q}, <)$, where $\mathbb{Q}$ is the set of all rational numbers and $<$ is the usual order. We can use this flow when we view time as dense; density may correspond more closely than discreteness to our natural intuition about time as we move through it. Density may also be useful in modelling distributed computing applications, in which a program may find another acting in between its own execution steps. By Cantor's theorem, $(\mathbb{Q}, <)$ is up to isomorphism the unique countable linear dense flow of time without endpoints.

   We may wish to impose a 'real-world' constraint on valuations into $(\mathbb{Q}, <)$. If atoms represent basic states of a system, we may decide that only finitely many changes in state may occur, either in any bounded interval, or at all. So we may restrict to temporal structures $(\mathbb{Q}, <, h)$ in which each atom may change its value only finitely often in any bounded interval, or at all.

4. The reals, $(\mathbb{R}, <)$ where $\mathbb{R}$ is the set of all real numbers, and $<$ is the usual ordering. Here we have not only density, but continuity and separability too. In fact, $(\mathbb{R}, <)$ is up to isomorphism the unique continuous separable linear flow of time without endpoints. The reals are one of the most interesting and expressive linear flows; most other common linear flows can be 'encoded' in them.

   Various restrictions on atoms can be considered in the context of $(\mathbb{R}, <)$. Bounded or finite variability is again a possibility. Another is to require that the values of assignments are 'simple' in some way. For example, we might restrict to temporal structures $(\mathbb{R}, <, h)$ with assignments $h : L \to \wp(\mathbb{R})$ such that for each $q \in L$, $h(q)$ is F_σ (a countable union of closed sets), or Borel (in the countably complete Boolean subalgebra of $(\wp(\mathbb{R}), \cup, \setminus, \emptyset, \mathbb{R})$ generated by the closed sets). See [33] for applications of this idea to decidability of temporal logics over $(\mathbb{R}, <)$.

5. The binary tree $\mathcal{T} = ({}^{<\omega}2, <)$. For ordinals α, β, we write ${}^{\beta}\alpha$ for the set of all maps $f : \beta \to \alpha$. We write ${}^{<\beta}\alpha$ for $\bigcup_{\gamma<\beta} {}^{\gamma}\alpha$. So ${}^{<\omega}2$ is the set of all maps from a natural number n into 2. (We treat natural numbers n as ordinals, so that $n = \{0, 1, \ldots, n-1\}$ and $2 = \{0, 1\}$.) We can regard such a map as a sequence of 0s and 1s of length n, so that ${}^{<\omega}2$ can be viewed as the set of all finite sequences of 0s and 1s. The ordering $<$ on $\mathcal{T}$ is that of proper initial segment: so $t < u$ iff u is a proper extension of t. Clearly, $\mathcal{T}$ is a tree. Each branch of $\mathcal{T}$, ordered by the restriction of $<$, is isomorphic to $(\mathbb{N}, <)$; trees with this property will be called *ω-trees.*

   In this case, we may sometimes wish to restrict to assignments $h : L \to \wp({}^{<\omega}2)$ such that $h(q)$ is finite for all $q \in L$. Topological restrictions can also be made: see, e.g., [85].

6. Relativistic temporal logic has been considered a little in the literature: see, e.g., [71] (reprinted in [73]) and [180]. For $n \geq 2$, we can define n-dimensional space-time $\mathcal{T}^n$ to be $(\mathbb{R}^n, \leq)$. Here, $\mathbb{R}^n$ is the set of all n-tuples of real numbers; the first $n-1$ coordinates are for space and the last is for time. We define $(x_1, \ldots, x_n) \leq (y_1, \ldots, y_n)$ iff $\sum_{i=1}^{n-1}(y_i - x_i)^2 \leq (y_n - x_n)^2$ and $x_n \leq y_n$. This ordering is reflexive (so strictly, we are not dealing with a flow of time), and transitive (exercise). $\mathcal{T}^4$ is the traditional Minkowski space-time.

2.2 Cycles

We have defined flows of time to be transitive. One context in which this assumption seems unjustified is in the occasionally studied case of circular or cyclical time. Interest in this has stemmed from studies of general relativity [70] as well as from philosophical and religious motivations. The intuitive idea here is that there is a sequence, or locally linear arrangement of time which has its start temporally after its end time point. If we follow along the flow of time through such a structure we see the same arrangements of atoms repeated forever (and into the past as well as into the future). Obviously this behaviour can be mimicked by making restrictions on the valuations of atoms on an appropriate transitive linear flow of time so that the same pattern of truth values is repeated periodically. Cyclical time models, however, are instead based on a cyclical arrangement of time points.

A moment's thought makes it clear that it is not straightforward to make a mathematical structure to model this idea. It is certainly inadequate to have a flow of time $(T, <)$ with a transitive relation $<$: each time point would be after every other time point as well as being after itself.

Two solutions to this have been proposed. In [104] we see the idea of a model of time (T, ϕ) based on a ternary between-ness relation ϕ. We say that y lies between x and z and put $\phi(x, y, z)$ iff moving forwards in time from x to y does not pass through z. Appropriate conditions on ϕ are suggested in [104].

In [158] we see instead the usual binary $<$ being used as an irreflexive, anti-symmetric and non-transitive relation with $x < y$ meaning something like that y is a while after x but not so long after, not more than half way round the cycle of time. According to this approach we have the following definition.

DEFINITION 6. A *cyclical* flow of time is $(T, <)$ with $<$ being a binary relation on the set T such that $<$ satisfies the following axioms:

total order:	$\forall xy[(x < y) \vee (x = y) \vee (y < x)]$
anti-symmetry:	$\forall xy \neg[(x < y) \wedge (y < x)]$
future transitivity:	$\forall xyzu[(x < y) \wedge (x < z) \wedge (x < u) \wedge (y < z) \wedge (z < u) \rightarrow (y < u)]$
past transitivity:	$\forall xyzu[(y < x) \wedge (z < x) \wedge (u < x) \wedge (y < z) \wedge (z < u) \rightarrow (y < u)]$
non-transitivity:	$\exists xyz[(x < y) \wedge (y < z) \wedge (z < x)]$

2.3 Branching Time

The term *Branching Time* is sometimes used for general, i.e. transitive and irreflexive but not necessarily linear flows of time.

However, there is a more specific use that is more common both in philosophical and in computing contexts. These are the flows of time $(T, <)$ which are trees according to definition 5 above, i.e. in which $<$ is linear towards the past. The branching towards the future is often used to capture the indeterminacy of the future: openness, choices and chances. The linearity of the past instead captures the fixed, (already) determined nature of that part of time.

We shall use *Branching Time* in this latter sense, i.e. to refer to flows of time which are trees. We will see that there are branching time temporal logics in which formulas are evaluated at points (sometimes called nodes) on such tree structures. Branches are maximally linearly ordered subsets of the the set of time points: see Definition 5. Note, however, that the terms *path*, or *history* are sometimes used instead of 'branch' in the literature. Also, it is sometimes the case that a *branch* may be a linearly ordered set of points maximally towards the future but not necessarily towards the past, i.e. 1) b is linear, 2) if $x \leq y$ for all $x \in b$, then $y \in b$, and 3) for all x, y, z, if $x < y < z$ and $x \in b$ and $z \in b$ then $y \in b$.

In Burgess's [28], a more complex branching time temporal structure is suggested. Burgess notes that in some situations it is useful to pick out certain branches of a tree structure as being legitimate and to ignore others. For example, in a computing application some branches might exhibit a required fairness property in that certain atoms are true an infinite number of times along the branch, while other branches are unfair and thus may not be considered to be able to eventuate. In philosophical applications

some branches may correspond to "intended" or "possible" branches while others may be deemed impossible. As discussed in [165], there are also technical reasons to consider tree structures with a pre-identified set of branches. See also [218, 219].

In order to support such reasoning, certain temporal logics are defined on tree structures with identified sets of branches. In order for all the points of the tree to be non redundant and to be able to play a role in the semantics of such logics, it is useful to suppose that for each point there is one of the identified branches containing that point. We thus have the following definition.

DEFINITION 7. We say that a set B of branches of a tree $(T, <)$ is a *bundle* (on $(T, <)$) iff for all $x \in T$ there is $b \in B$ such that $x \in b$. If B is a bundle on tree $(T, <)$ then we say that $(T, <, B)$ is a *bundled tree (frame)* and, if h is a valuation, then $(T, <, B, h)$ is a *bundled structure.*

Temporal logics can be defined on bundled tree structures. Of course, the bundle only comes in to play in the semantics if the logic allows some sort of quantification over branches. Bundled tree temporal logics will also be appropriate in the case that a bundle B in a bundled tree structure is the set $B(T, <)$ of all branches of the tree $(T, <)$. In that case we say that the bundle is *complete.* Bundled tree temporal logics can thus also be used on plain, not bundled tree structures: just use the complete bundle as a default.

Major variations in branching time temporal logics on tree structures, plain or bundled, arise from differences in the locations of evaluation of formulas. If we consider truth to reside in points then we evaluate formulas at points (or worlds) in tree structures. An eminent tradition, however, requires that truth is evaluated at points on branches in structures, i.e. evaluation is at a 'world' = pair (time, history). We will investigate the differences more fully later when we consider the syntax and semantics of temporal logics which utilise them.

This choice of location of evaluation does, however, give rise to an important but subtle difference in structures. Those structures which use time-branch pairs for evaluation of formulas may also permit valuations of atoms to be sensitive to branch. Thus the atom p may evaluate to "true" on branch b at point x but evaluate to "false" on branch b' at point x.

DEFINITION 8. Suppose that $(T, <)$ is a tree.

A map $h : L \to \wp(T)$ is called a *local* assignment.

A map $h : L \to \wp\{(b, x) \mid x \in b \in B(T, <)\}$ is called a *non-local* assignment.

We use similar definitions for the case of bundled structures.

Local versus non-local assignments were first distinguished by Prior in [150]. In [218], they are discussed in terms of atoms containing (or not) a trace of "futurity". Should the truth of an atom at a time, now say, be dependent on which of the many possible futures actually comes about after now? Structures with a non-local assignment to atoms allow atoms to contain this trace of futurity. An example of Stefan Wölfl (quoted in [219]) is the seemingly atomic statement "the King is dying", the truth of which actually depends on the branch of evaluation.

The evaluation of atoms and/or more general formulas at time-branch pairs suggests an alternative model of time which allows us to keep the more traditional modal logic approach of evaluating formulas at worlds. We can arrange the time-branch pairs in a two-dimensional structure. Such an arrangement was suggested by Kamp and recorded in [189].

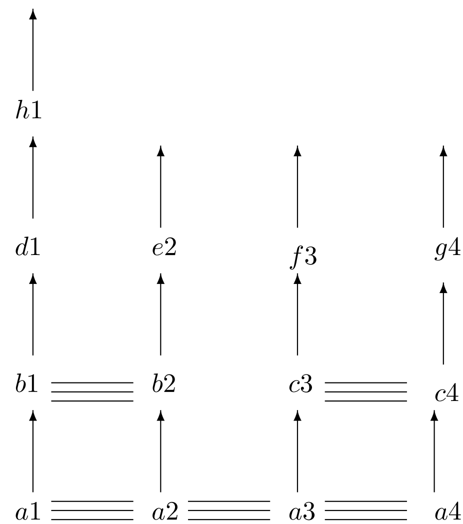


Figure 1. A Kamp Frame

DEFINITION 9. A *Kamp frame* is a triple $(K, <, \equiv)$, where:

1) K is the set of points;
2) $<$ is a union of linear orders on K
   i.e., $\forall xyz(x < y \wedge y < z \to x < z)$,
   $\forall xy\neg(x < y \wedge y < x)$,
   $\forall xyz(x < y \wedge x < z \to (y < z \vee y = z \vee z < y))$
   and $\forall xyz(y < x \wedge z < x \to (y < z \vee y = z \vee z < y))$;
3) $\equiv$ is an equivalence relation such that for all $x, y \in K$:
   if $x \equiv y$ then we do not have $x < y$,
   if $x \equiv y$ and $u < x$ then there is $v < y$ such that $u \equiv v$, and
   if $x \equiv y$ and for all $u > x$ there is $v > y$ with $u \equiv v$ then $x = y$.

A Kamp-structure is a Kamp Frame with a valuation of the atoms which agrees across $\equiv$. Corresponding to each Kamp-structure is a bundled structure: see Figures 1 and 2. The $<$ relation relates worlds (i.e. time-branch pairs) on the same branch while the $\equiv$ relation relates worlds which represent the same time point possibly paired with different branches. See [165] for details.

The special case of ω-trees, i.e. those in which each branch is isomorphic to the natural numbers, is of particular interest in computing. These structures arise as representations of the possible runs starting from a fixed state through a transition system. Let us be more precise.

DEFINITION 10. A *total frame* is a pair (S, R), where:

- S is the non-empty set of *states*
- R is a total binary relation $\subseteq S \times S$
  (i.e. for every $s \in S$, there is some $t \in S$ such that $(s, t) \in R$)

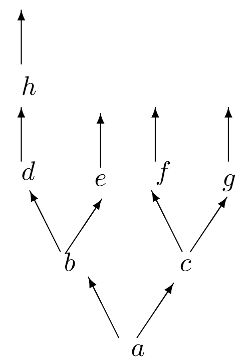


Figure 2. The corresponding tree

DEFINITION 11. A *transition system* is a triple $M = (S, R, g)$ where:
(S, R) is a total frame,
$g : S \to \wp\mathcal{L}$ is a labelling of the states with sets of atoms.

A *fullpath* (or branch or run) in M (or in (S, R)) is an infinite sequence $s_0, s_1, s_2, \ldots$ of states of M such that for each i, $(s_i, s_{i+1}) \in R$. For the fullpath $b = s_0, s_1, s_2, \ldots$, and any $i \geq 0$, we write b_i for the state s_i, $b_{\leq i}$ for the prefix sequence $s_0, s_1, \ldots, s_i$ and $b_{\geq i}$ for the fullpath $s_i, s_{i+1}, s_{i+2}, \ldots$.

The set of all prefixes of fullpaths of a transition system M with a fixed initial state naturally forms a tree: we put $s_0, s_1, \ldots, s_i \leq r_0, r_1, \ldots, r_j$ iff $i \leq j$ and for all $k \leq i$, $s_k = r_k$. The branches of this tree, sometimes called the unwrapping of the transition system, correspond exactly to the fullpaths themselves.

We can easily put an assignment to the atoms on the unwrapping of a transition system. Put $h(p) = \{b_{\leq i} \mid p \in g(b_i)\}$.

For a deeper survey of the range of models used for branching-time semantics, the reader is referred to [218, 219]. *Alternating time* generalises branching time in some ways, and was considered in [9, 78].

2.4 Granularity

For certain applications of temporal reasoning it is important to allow formulas which refer to behaviour occurring across a combination of different layers of time measurements. For example, we might want to say that a property holds during the last week of every month, thus supposing that time is divided somehow into weeks as well as somehow into months. This multi-layered multi-grained aspect of temporal experience is known as *granularity* and several formal structures have been suggested and investigated in order to represent this. One basic suggestion is to use trees in which each particular height of nodes represents a given granularity: a node at that height corresponds to a one unit period of time at that granularity, the nodes at that height are ordered by the earlier-later relation and the children nodes in the tree represent the periods of time at the next finest granularity which comprise that period. For example, a node representing a week may have 7 children representing its days.

For more on granularity, we refer the reader to references such as [55], [137], [17].

2.5 Many-dimensional Evaluation

Some temporal expressions may involve what seems to be two or more dimensions of time. Examples here include formalizations of natural language expressions which rely on multiple points of evaluation (for example see [155, 113]), properties which hold over intervals of time and circumstances where time is being used in several different ways. In such examples there may be a fundamental choice of how to construct a formal temporal model to capture the situation. We might be able to use a standard linear one-dimensional flow of time as defined above, but on which the truth of expressions are evaluated at pairs (or tuples) of time points: we discuss this approach briefly when we consider syntax and evaluation in the next section.

The alternative approach is to model the situation using structures which themselves have a multi-dimensional aspect. We have already considered branching structures which may be regarded, especially in their Kamp-frame incarnation, as two-dimensional. Shortly we will look at interval structures and these are sometimes built from two dimensions of time.

Another important example is in reasoning about databases and database management systems, particularly regarding temporal databases which need to record some timing or date information. In temporal database work, a distinction is sometimes made between *valid time,* that is the date-time when an event or state holds in the world, and *transaction time,* that is when a fact or property is recorded in the database. In reasoning about the correctness of a database system it may be important to have explicit descriptions about relationships between valid and transaction times of various events. See [188], [49] for example. To provide formal semantics for such reasoning, it is useful to consider structures which are a cartesian product of a flow of valid time and a flow of transaction time.

2.6 Intervals

As we will see below, it is possible to reason about linear structures, such as those we have met above, by considering descriptions of the behaviour of atoms across intervals of time. This reasoning is about point-based temporal structures: the atoms are evaluated at points. From these atoms, more complicated expressions can be built which need to be evaluated on intervals of time.

An alternative approach to interval temporal logics is to evaluate propositional atoms on intervals of time. There are natural language motivations for doing this: for example, it is natural to explain the semantics of the proposition "I climbed Everest" with reference to an interval of time. Intervals of time can be posited as the basic ontological entities. An interval temporal structure could be a set of intervals from a linear structure along with an evaluation for the atoms on the intervals.

In general, an interval could be defined to be any convex set of points from a linear structure: i.e. I is an interval of $(T,<)$ iff for all $x < y$ from I, for all z from T, if $x < z < y$ then $z \in I$. Different structures can arise from the choice of $(T,<)$ (e.g., the integers, or the reals) or by only looking at some subset of the set of all intervals, for example the set of closed intervals of the reals.

Allen identified 13 basic relations that may hold between intervals. One interval may precede another with a gap in between, or it may just end where the second one starts,

etc. See [6] for details. An algebra can be defined to capture the relationships between these relations.

The approach of *taking intervals as primitive objects* has also received much attention [192, 87, 196]. A structure is then typically a set I of abstract objects thought of as 'intervals', together with some relations on them, perhaps motivated by some of Allen's 13 relations, and an assignment of atoms to sets of intervals, just as in the point-based approach we assign atoms to sets of time points. For example, van Benthem in [192] takes as primitive relations '$i < j$' and '$i \sqsubseteq j$', and their converses. The intuition is that $i < j$ means that every point in i is earlier than every point in j, and $i \sqsubseteq j$ means that all points in i are in j; but these meanings are only analogies, since i, j are not necessarily intervals but only abstractions of intervals. The 'interval structures' then have the form $(I, <, >, \sqsubseteq, \sqsupseteq, h)$ where $h : L \to \wp(I)$. The elements of I are not necessarily sets of time points, and the interval structure may not even be isomorphic to a set of intervals arising from time points, with relations induced from the point structure. However, it may be of interest to characterise when there is such an isomorphism or to consider similar representation results [119, 7, 196, 31, 20, 76].

2.7 Combining temporal logic with other logics

So far in this section, we have been presenting structures that will serve as semantics for *propositional* logics of time. They have had the form (M, h), where M consists of a set of objects — points, intervals, etc., — representing time, with additional structure such as an earlier-later relation, and h assigns truth values to atoms at time objects in M. Thus, we envisage that the atomic formulas of the associated logics (to be developed in the next section) will be essentially atoms.

We now wish to examine how to handle more complicated atomic formulas. Their meaning will not simply depend on a time object, but also on extra, typically *non-temporal,* information. Since the non-temporal domain will typically have its own associated logic, we are in the area of *combining logics,* or *many-dimensional logics,* to be discussed in detail in Chapter 15. See also [131]. There are applications in computing, philosophy, linguistics, etc., in which these more expressive logics can be useful; they also present intriguing mathematical problems that have been more and more actively studied in the last few years.

Here, we discuss three kinds of structure that offer semantics for combinations of temporal logic with other logics.

1. *Epistemic temporal logic* (see Chapter 18 and [88]) combines temporal logic with a logic of knowledge. To model knowledge, we can use a Kripke frame (W, R). W is the set of possible worlds, and each $w \in W$ is regarded as a state of knowledge. The accessibility relation R is set up to reflect the meaning of $\Box A$ as 'A is known'. To model *temporally-dependent knowledge,* we can use structures that have the form $(T, <, W, R, h)$, where $(T, <)$ is a flow of time, (W, R) is a Kripke frame suitable for representing knowledge, and $h : L \to \wp(T \times W)$ is an assignment: a pair $(t, w) \in h(q)$ is one in which q is regarded as true at time t in state w. Both temporal and epistemic operators can be given meaning in such structures.

2. This approach can be taken further, by adding in a similar way a temporal dimension to an arbitrary Kripke frame. For example, if we use an S5 frame on which

the accessibility relation is global, we obtain essentially parallel identical flows of time, indexed by the worlds of the frame. If we use an equivalence relation that is refined as we move through time, so that any two points equivalent at time t were also equivalent at all earlier times than t, we obtain a special kind of *Kamp frame,* as in definition 9. If we use another flow of time (or another copy of the same flow) then we obtain a structure suitable for two-dimensional temporal logic as described above.

3. One of the most interesting combinations is of temporal logic with classical first-order logic, to obtain *first-order temporal logic.* There are very many options over exactly how to do it: see Chapter 9 and also Garson's excellent survey [68] on first-order modal logic. We confine ourselves here to discussing a single, fairly powerful option, summarised as *constant domains, rigid constants, and flexible functions and predicates.*

   The atomic formulas of our logic will be classical atomic first-order formulas. So we fix a signature (or similarity type, or vocabulary) L, consisting of relation symbols, function symbols, and constants (in many treatments, function symbols and even constants are omitted). A temporal structure will have the form $\mathcal{M} = (T, <, D, (M_t : t \in T))$, where $(T, <)$ is a flow of time, D is a non-empty set (the *domain* of $\mathcal{M}$), and for each $t \in T$, M_t is an L-structure with domain D (so the domain is constant over time). Let us write the interpretation in M_t of a symbol $s \in L$ as s^{M_t}. We require that $c^{M_t} = c^{M_u}$ for each constant $c \in L$ and all $t, u \in T$ (so constants are 'rigid', their meanings not varying over time). There are no restrictions on the interpretations of relation and function symbols; these may vary over time, so are called 'flexible'. For some purposes, for example in databases, we may wish to restrict consideration to models with finite domains.

3 TEMPORAL LOGICAL SYSTEMS

In the preceding section, we set out a variety of temporal structures. Now we have to set up some logical syntax for expressing properties of them. There are many choices here. The most basic choice is whether to adopt an external or internal perspective. Some logical systems, such as first-order logic, make statements about a temporal structure 'from the outside': sentences of these logics are evaluated relative to an entire temporal structure. They are natural for making statements involving 'before' or 'after'. There are other, more modal-style systems in which formulas are evaluated 'internally': e.g., at individual time points. They are more natural for expressing tense constructions like 'tomorrow', 'used to be', and so on. There is an extensive philosophical literature on this point (see, e.g., [149, 150, 151]). The distinction between the two approaches is somewhat artificial, in that each can simulate the other to some extent. But the choice of whether to use first-order logic or extensions of it, or to adopt a more modal-style logic, is still a major decision point.

Modal-style logics of time — commonly referred to as *temporal logics* — have received much attention. Several kinds of temporal logic are available, of varying expressivity, so one may hope to choose the best logic for the problem in hand. Temporal logics are well understood, and compared with first-order-based systems, they are arguably closer to

natural language, and reasoning with them is generally computationally simpler. Various computer implementations of temporal logics exist and have been very successful.

But in fact, it pays to take both the internal and external approaches seriously. There is no real reason not to: each is a legitimate way to handle time. Some situations lend themselves to one, some to the other. A key point is that the availability of both kinds of system allows us to compare them — for example, in expressive power. Roughly, though in some situations they are equally expressive, in general we find that the first-order approach is stronger and more succinct, and this is another reason to consider it. Though the computational complexity of first-order systems is usually higher than for modal-style ones, still there exist powerful computer theorem-provers for first-order logic. First-order logic and its extensions can serve as a benchmark for expressivity of modal-style temporal logics. Moreover, results in first-order logic can be transferred to temporal logic, and potentially vice versa.

We begin this section by exploring some first-order-based systems for time. Afterwards, we set out some 'modal-style' temporal logics. Then we consider extensions of basic temporal logic with various second-order operations. Finally, we examine some logical syntaxes with operations specific to branching time.

3.1 First-order logic

Consider temporal structures of the form $\mathcal{M} = (T, <, h)$, where $(T, <)$ is a flow of time and $h : L \to \wp(T)$ is an assignment, L being the underlying set of atoms. The natural first-order logic appropriate for such structures has signature

$$L^\star = \{<\} \cup \{Q : q \in L\},$$

where $<$ is a binary relation symbol, and for each atom $q \in L$, Q is a unary relation symbol. We may regard $\mathcal{M}$ naturally as an L-structure $\mathcal{M}^\star$, defined as follows. The domain of $\mathcal{M}^\star$ is T. '$<$' is interpreted as the given earlier-later relation $<$ on T. Finally, for each relation symbol Q $(q \in L)$, we set $\mathcal{M}^\star \models Q(t)$ iff $t \in h(q)$, for all $t \in T$. We may now write whatever first-order L-formulas we wish, and evaluate them in $\mathcal{M}^\star$. For example, $\forall x \exists y (y > x)$ is true in $\mathcal{M}^\star$ just when the flow of time $(T, <)$ has no last moment. If the flow of time is $(\mathbb{N}, <)$, we can express that an atom q is true infinitely often, by $\forall x \exists y (y > x \wedge Q(y))$.

3.2 Monadic second-order logic

Second-order formulas can also be useful. *Monadic second-order logic* has a prominent role in the logic of time, as we shall see. In this logic, we can quantify over both individual time points and sets of time points (i.e., unary relations on time), but not over binary and higher-order relations. The syntax of monadic second-order logic is as follows. We have a set $\mathcal{V}_1$ of first-order variables, which will be written $x, y, \ldots$, and a set $\mathcal{V}_2$ of second-order variables, written $X, Y, \ldots$. Both these sets are generally taken to be countably infinite. The atomic formulas of our system are $x = y$, $x < y$, $Q(x)$ for each $q \in L$, and $X(x)$; and if φ, ψ are monadic second-order formulas, then so are $\neg\varphi$, $\varphi \wedge \psi$, $\exists x \varphi$, and $\exists X \varphi$ — all for each $x, y \in \mathcal{V}_1$ and $X \in \mathcal{V}_2$. Evaluation of formulas takes place in a temporal structure $\mathcal{M} = (T, <, h)$ with respect to assignments $\nu_1 : \mathcal{V}_1 \to T$ and $\nu_2 : \mathcal{V}_2 \to \wp(T)$. We define $\mathcal{M}, \nu_1, \nu_2 \models \varphi$ by induction on φ:

- $\mathcal{M}, \nu_1, \nu_2 \models x = y$ iff $\nu_1(x) = \nu_1(y)$,
- $\mathcal{M}, \nu_1, \nu_2 \models x < y$ iff $\nu_1(x) < \nu_1(y)$,
- $\mathcal{M}, \nu_1, \nu_2 \models Q(x)$ iff $\nu_1(x) \in h(q)$,
- $\mathcal{M}, \nu_1, \nu_2 \models X(x)$ iff $\nu_1(x) \in \nu_2(X)$,
- $\mathcal{M}, \nu_1, \nu_2 \models \neg\varphi$ iff $\mathcal{M}, \nu_1, \nu_2 \not\models \varphi$,
- $\mathcal{M}, \nu_1, \nu_2 \models \varphi \wedge \psi$ iff $\mathcal{M}, \nu_1, \nu_2 \models \varphi$ and $\mathcal{M}, \nu_1, \nu_2 \models \psi$,
- $\mathcal{M}, \nu_1, \nu_2 \models \exists x\varphi$ iff $\mathcal{M}, \nu_1', \nu_2 \models \varphi$ for some $\nu_1' : \mathcal{V}_1 \to T$ with $\nu_1'(y) = \nu_1(y)$ for all $y \in \mathcal{V}_1 \setminus \{x\}$,
- $\mathcal{M}, \nu_1, \nu_2 \models \exists X\varphi$ iff $\mathcal{M}, \nu_1, \nu_2' \models \varphi$ for some $\nu_2' : \mathcal{V}_2 \to \wp(T)$ with $\nu_2'(Y) = \nu_2(Y)$ for all $Y \in \mathcal{V}_2 \setminus \{X\}$.

This is a very expressive logic. For example, over the natural numbers $(\mathbb{N}, <)$, we can express that an atom q, represented by the unary relation symbol Q as usual, is true only at even numbers:

$$\exists X(X(0) \wedge \forall xy[x < y \wedge \neg\exists z(x < z < y) \to (X(x) \leftrightarrow \neg X(y))] \wedge \forall x(Q(x) \to X(x))).$$

This cannot be expressed in first-order logic.

3.3 *Temporalised first-order logic*

When we are dealing with a combination of time with first-order logic, a *two-sorted first-order* approach can be used. Recall from Section 2.7 that in this setting, L is an ordinary first-order signature, and structures have the form $\mathcal{M} = (T, <, D, (M_t : t \in T))$, where $(T, <)$ is a flow of time and each M_t ($t \in T$) is an L-structure with domain D. Constants are interpreted rigidly (each constant has the same interpretation in each M_t).

To handle this in two-sorted first-order logic, we use sorts $\mathbf{d}$, $\mathbf{t}$ (standing for 'domain' and 'time', respectively). We introduce an $(n+1)$-ary relation symbol $R^\star$ of sort $\mathbf{t} \times \mathbf{d}^n$, for each n-ary relation symbol $R \in L$. (The notation $\mathbf{t} \times \mathbf{d}^n$ means that the first argument of $R^\star$ is of sort $\mathbf{t}$ and the last n arguments are of sort $\mathbf{d}$.) That is, we make our relation symbols time-dependent. The same process is undertaken for function symbols in L. Each constant symbol of L is rigid in $\mathcal{M}$, so we simply make a copy of it, of sort $\mathbf{d}$. We let $L^\star$ be a new signature consisting of all these symbols. $L^\star$ also has a binary relation symbol $<$ of sort $\mathbf{t} \times \mathbf{t}$.

We now define a two-sorted $L^\star$-structure $\mathcal{M}^\star$ from $\mathcal{M}$. The domain of $\mathcal{M}^\star$ is the disjoint union of two sets, D (sort $\mathbf{d}$) and T (sort $\mathbf{t}$). For n-ary $R \in L$, $a_1, \ldots, a_n \in D$, and $t \in T$, we define $\mathcal{M}^\star \models R^\star(t, a_1, \ldots, a_n)$ iff $M_t \models R(a_1, \ldots, a_n)$. For an n-ary function symbol $f \in L$, $a_1, \ldots, a_n, b \in D$, and $t \in T$, we let $\mathcal{M}^\star \models f^\star(t, a_1, \ldots, a_n) = b$ iff $M_t \models f(a_1, \ldots, a_n) = b$. We let $c^{\mathcal{M}^\star} = c^{M_t} \in D$, for each constant $c \in L$ and any $t \in T$; this is well-defined since constants are rigid in $\mathcal{M}$. And $<$ is interpreted in $\mathcal{M}^\star$ as the earlier-later relation $<$ from $(T, <)$. Now we can readily write two-sorted $L^\star$-formulas to describe the original $\mathcal{M}$, via $\mathcal{M}^\star$. For example, if L contains unary relation symbols dog and day to pick out the dogs and days in an L-structure $\mathcal{M}$, then the $\mathcal{L}^\star$-sentence $\forall t, x(\mathsf{dog}^\star(t, x) \to \exists t\ \mathsf{day}^\star(t, x))$ is true in $\mathcal{M}^\star$ just in case every dog has his day.

3.4 Temporal Logics

Let us now examine the 'modal-style' temporal logics.

The simplest is just the propositional modal logic K4 which has a modal diamond $\Diamond$ added recursively to the formulas of classical propositional logic. The formulas are just then $p \in L$, $\neg\alpha$, $\alpha \wedge \beta$, and $\Diamond\alpha$ where α and β are themselves formulas. The logic can be presented in a semantic way by evaluating these formulas on structures $(T, <, h)$ in which $(T, <)$ is a transitive irreflexive flow of time.

To be absolutely clear in this first case we present the full semantics. Formulas are evaluated at points of time $t \in T$ in such structures $\mathcal{T} = (T, <, h)$. We write $\mathcal{T}, t \models \alpha$ to represent α being true at time point t in the structure. This is defined formally recursively as follows:

$\mathcal{T}, t \models p$ iff $t \in h(p)$,
$\mathcal{T}, t \models \neg\alpha$ iff $\mathcal{T}, t \not\models \alpha$,
$\mathcal{T}, t \models \alpha \wedge \beta$ iff $\mathcal{T}, t \models \alpha$ and $\mathcal{T}, t \models \beta$,
$\mathcal{T}, t \models \Diamond\alpha$ iff there exists $s \in T$ such that $t < s$ and $\mathcal{T}, s \models \alpha$.

K4 is a traditional modal logic. The simplest seriously temporal logic incorporates a modal diamond directed towards the past as well as one directed towards the future. This is one of the temporal logics invented by the "father" of temporal logic, Arthur Prior. In order to present this clearly we switch to his original notation of using F for the future diamond and P for the past diamond. The syntax is obtained by adding both the following two clauses to the standard ones for classical propositional logic: if α is a formula of the logic then so are $F\alpha$ and $P\alpha$. The logic can be given semantics on any temporal structure. The additional semantic clauses are:

$\mathcal{T}, t \models F\alpha$ iff there exists $s \in T$ such that $t < s$ and $\mathcal{T}, s \models \alpha$ (as above in K4)
$\mathcal{T}, t \models P\alpha$ iff there exists $s \in T$ such that $s < t$ and $\mathcal{T}, s \models \alpha$

The dual, modal box, versions of F and P are traditionally known as G and H. $G\alpha$ is true at a time point in a structure if and only if α is going to be true at all points in the future of that point. G can be introduced as a temporal connective in its own right, like F or P, or it can be just regarded as an abbreviation: $G\alpha \equiv \neg F \neg \alpha$. Likewise for H, "has always been true".

Temporal languages with some combination of these connectives F, P, G and H can be given semantics over the class of all temporal structures (i.e. flows of time with assignments to the atoms). Equally, they can be given semantics on smaller classes of structures, even on the set of structures with a given fixed flow of time. Making such restrictions generally gives rise to different temporal logics in the sense of the set of valid formulas of the logic. Take Prior's propositional language with F and P. The formula $Fp \wedge Fq \rightarrow (F(p \wedge Fq) \vee F(q \wedge Fp) \vee F(p \wedge q))$ is not valid: there are structures in which there are points at which this formula is false. If however, we restrict attention to structures over linear flows of time then this formula is valid. We discuss validity and related issues in a later section.

We have used the term "temporal connective" loosely above. It is time to make this a little more precise. By a temporal connective we will mean a tuple consisting of a symbol, its arity and a semantic clause. For example we have seen the 1-ary connective F with its future directed clause. Below we will see how to present the semantic clause in a standard way. This will allow temporal logics to be constructed by choosing a base logic, say classical propositional or predicate logic, choosing a set of connectives, and choosing a class of structures on which the semantics is evaluated. Such flexibility is a

very powerful feature of temporal logics: the right logic can be put together for a specific application.

Before doing this let us mention a few variations on Prior's original temporal connectives. When these connectives are used for computing applications and when the flow of time is taken to be the natural numbers then we often see a variant of F, commonly written as $\Diamond$, which has a reflexive semantics. We might use the symbol $F_{\leq}$ for this "non-strict" version of F. The semantic clause is:

$\mathcal{T}, t \models F_{\leq}\alpha$ iff there exists $s \in T$ such that $t \leq s$ and $\mathcal{T}, s \models \alpha$.

There are similar non-strict versions of P, G and H. Note too, that in general there is no reason for them only to be seen in the context of natural numbers time.

Also in the context of natural numbers time, and especially in combination with non-strict F, it is common to see a connective written as X, T, or $\bigcirc$, and meaning 'tomorrow' or 'next-time'. The semantics could be presented for general flows as

$\mathcal{T}, t \models X\alpha$ iff there exists $s \in T$ such that $t < s$ and $\mathcal{T}, s \models \alpha$ and there is no u with $t < u < s$,

but for natural numbers flows only, we can be more straightforward and define

$(\mathbb{N}, <, h), n \models X\alpha$ iff $(\mathbb{N}, < h), n+1 \models \alpha$.

A yesterday or last-time connective can also be given but there is a subtlety: what to do at the start of time? In fact, there end up being two yesterday connectives, a weak yesterday W and a strong yesterday Y:

$(\mathbb{N}, <, h), n \models W\alpha$ iff $n = 0$ or $(\mathbb{N}, < h), n-1 \models \alpha$, and

$(\mathbb{N}, <, h), n \models Y\alpha$ iff $n > 0$ and $(\mathbb{N}, < h), n-1 \models \alpha$.

These are in fact duals of each other: $W\alpha \equiv \neg Y \neg \alpha$. Tomorrow is self-dual.

Arbitrary temporal connectives with first-order tables. For an n-ary connective $\#$, there will be a *table* $\tau_{\#}(t, P_1, \ldots, P_n)$ for $\#$ being a formula of the first-order logic $L^{\star}$ (introduced in Section 3.1 above), written using variables including t, and the 1-ary relation symbols $P_1, \ldots, P_n$. The table gives the semantics of $\#$ in the following sense. Suppose that $\#$ is one of the temporal connectives in a temporal language so that the formulas include $\#(\alpha_1, \ldots, \alpha_n)$ whenever $\alpha_1, \ldots, \alpha_n$ are formulas. Suppose that $\mathcal{T} = (T, <, h)$ is a temporal structure. As usual we define truth of all formulas at all time points in $\mathcal{T}$ by induction on the construction of the formulas. The semantic clause for $\#(\alpha_1, \ldots, \alpha_n)$ is used when we have defined the truth of the α_i at every time point. First, let $S_i = \{b \in T \mid \mathcal{T}, b \models \alpha_i\}$. Then the semantic clause tells us that for any $a \in T$, $\mathcal{T}, a \models \#(\alpha_1, \ldots, \alpha_n)$ iff $\mathcal{T}^{\star} \models \tau_{\#}(a, S_1, \ldots, S_n)$ (where we use the semantics for $L^{\star}$ over $\mathcal{T}^{\star}$ defined in Section 3.1 above).

Some tables of connectives we have already met include:

Fp	$\exists s(t < s \wedge P(s))$
$F_{\leq}p$	$\exists s(t \leq s \wedge P(s))$
Pp	$\exists s(s < t \wedge P(s))$
Gp	$\forall s(t < s \rightarrow P(s))$
Hp	$\forall s(s < t \rightarrow P(s))$
Xp	$\exists s(t < s \wedge P(s) \wedge \neg \exists u(t < u < s))$
Yp	$\exists s(s < t \wedge P(s) \wedge \neg \exists u(s < u < t))$

In more philosophical temporal logic work, and especially that motivated by trying to give formal semantics to natural language tense constructs, a connective capturing the idea of "until" is often seen as $U(\alpha, \beta)$. Its table is:

$U(p, q)$ $\exists s\big((t < s) \wedge P(s) \wedge \forall u[((t < u) \wedge (u < s)) \rightarrow Q(u)]\big)$

This is read as "until p, q", meaning that until p holds we have q holding: q holds at all time points after now until some time at which p holds. There is a mirror image "since" connective. Its table is:

$S(p,q) \quad \exists s\big((s<t)\wedge P(s)\wedge\forall u[((s<u)\wedge(u<t))\rightarrow Q(u)]\big)$

In many computing applications an "until" connective is also seen. This is commonly written $\alpha U\beta$ and read as 'α holds until β does'. The semantics is:

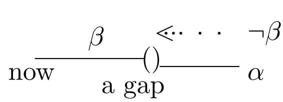
$pUq \quad \exists s\big((t\leq s)\wedge Q(s)\wedge\forall u[((t\leq u)\wedge(u<s))\rightarrow P(u)]\big)$

Thus, pUq is not the same as $U(q,p)$! The "philosophical" until is sometimes called the strict one (as $U(p,q)$ does not hold just because p holds now) while the "computing" until is called the non-strict one: qUp holds if p holds now. We will sometimes write $U_<$ for the former and $U_\leq$ for the latter: otherwise, they will be distinguished by the context. A mirror image $S_\geq$ of the S (or "$S_>$") above can be defined similarly.

For the most basic computing applications of temporal logic the much favoured language is the (propositional) temporal logic with the next-time X connective and the non-strict U connective. This language and/or its logic over the natural numbers flow of time is often called PLTL or PTL and it was introduced in the papers [147] and [66] which first proposed using temporal logics for reasoning about programs. It is easy to see that this logic is equally as expressive as the logic with just strict until over the natural numbers.

The language with X, $U_\leq$, Y, and $S_\geq$, sometimes called TL, was recommended for computing applications in [124] as the past-time connectives allow more natural expression of certain properties of interest to computer scientists. However, it is not hard to use the *separation* results mentioned in Section 4.6 to show that the past-time connectives are not really necessary in order to express any properties of a natural numbers-flowed structure (as far as evaluating properties at the zero time is concerned).

Expressivity concerns also led to the introduction of certain more complicated connectives for use with dense, in particular non-Dedekind-complete, flows of time. Such flows of time may have "gaps" or Dedekind cuts: that is, the flow can be partitioned into two non-empty sets A and B, say, with every time in A before every time in B, but A does not have a last time and B does not have a first time. These connectives are the *Stavi connectives* U' and S' which were mentioned in [66] as having to be added to (strict) 'until' and 'since' to achieve expressive completeness over general linear time. $U'(\alpha,\beta)$ holds if β is true from now until a gap in time after which β is arbitrarily soon false but after which α is true for a while: $U'(\alpha,\beta)$ is as pictured



S' is defined via the mirror image. Despite involving a gap, U' is in fact a first-order connective. Here is the first-order table for U':

$$
\begin{aligned}
&U'(p_1,p_2)\equiv\\
&\quad\exists s\Big(t<s\wedge\exists u(t<u<s\wedge\neg P_2(u))\wedge\exists u[t<u<s\wedge\forall v(t<v<u\rightarrow P_2(v))]\\
&\qquad\quad\wedge\forall u\big(t<u<s\rightarrow\big[\exists v(u<v\wedge\forall w(t<w<v\rightarrow P_2(w)))\big]\\
&\qquad\qquad\qquad\qquad\vee\big[\forall v(u<v<s\rightarrow P_1(v))\wedge\exists v(t<v<u\wedge\neg P_2(v))\big]\big)\Big)
\end{aligned}
$$

Roughly, this says that there is some $s>t$ which is on the other (future) side of the gap, so that any u between t and s is either before the gap (first disjunct) or after the gap

(second disjunct). Also, p_2 is false before s and true for a while after t. Of course, S' has the mirror image table.

Note that if $(T, <)$ is Dedekind complete then any formula of the form $U'(\alpha, \beta)$ or $S'(\alpha, \beta)$ is everywhere false.

3.5 Extensions of temporal logic

Now we very briefly mention a few miscellaneous extensions to temporal logics.

1. *Hybrid logic.*

   It has long been an undertaking for temporal logicians to invent more expressive languages and one temptation is to import some of non-modal first-order logic's abilities to reason explicitly about the states, i.e. the time points. Examples include Prior's "third grade tense logics" [150], the universal quantifier in [26], the "holds" predicate for intervals in [7] and more recent work in [19] and [74]. Temporal logics which allow some sort of naming of points within formulas are called *hybrid* logics, and these are discussed fully in Chapter 14. Having names for time points is particularly helpful for temporal reasoning tasks and systems of temporal inference can be designed to exploit this.

2. *Metric temporal logic.*

   In many applications of complex systems, timing or metric considerations are important. Reasoning about the behaviour of safety critical systems [144] and multimedia specifications [24] are just two examples. A good account of this so-called real-time logic area appears in [8]. A formula of the logic TPTL [8],

   $$\Box x.(p \to \Diamond y.(q \wedge y \leq x + 1))$$

   for example, can express that every p-state is followed by a q-state in at most one unit of time. Most of the timing work is built on discrete time temporal logics (for example via the timed state sequence formalism of [8]) and indeed any move to a dense order of times usually results in highly undecidable logics [8]. An alternative approach with reasonable complexity and comparable expressiveness is suggested in [164] via coding of "ticks" of a timer into the temporal logic with 'until' and 'since' over real numbers time. Of course, using abbreviations in terms of ticks as above does not allow us to quantify over metric values. However, as pointed out in [8] it is just this facility which makes metric temporal logics undecidable. Some recent work in metric logics includes [91, 125, 23].

3. *Many-dimensional connectives.*

   So far, we have concentrated on temporal logics whose formulas are evaluated at single time points. They are analogous to first-order formulas with one free variable. But first-order formulas can have many free variables, and by the same token it is sometimes useful to consider temporal formulas that are evaluated at k time points, for any fixed finite $k \geq 1$. The first of these points is rather special, and is called the *evaluation point.* The remaining $k - 1$ extra points are analogous to so-called *reference points* in natural language.

For semantics, we use a flow of time as usual, but there is a choice about how to evaluate atoms in it. Allowing an atom's value to depend on k time points leads us to true many-dimensional temporal logic, and to the interval logics discussed below. Such logics are often computationally intractable, and anyway it is often better to treat them as being ordinary one-dimensional evaluation but in a many-dimensional structure (cf. Section 2.5, and also Chapter 15).

Another option is to allow the values of atoms to depend only on the evaluation point, not the reference points. In consequence, a temporal structure remains of the form $\mathcal{T} = (T, <, h)$, where $h : L \to \wp(T)$. To formalise the semantics of an n-ary connective $\#$, we use a table of the form $\tau_{\#}(x_1, \ldots, x_k, P_1, \ldots, P_n)$, where $P_1, \ldots,$ P_n are now k-ary relation symbols. For time points $t_1, \ldots, t_k \in T$, we define $\mathcal{T}, t_1,$ $\ldots, t_k \models \alpha$ by induction on formulas α. For atomic α, we set $\mathcal{T}, t_1, \ldots, t_k \models \alpha$ iff $t_1 \in h(\alpha)$. The booleans are as expected. Finally, if $\alpha_1, \ldots, \alpha_n$ are formulas and inductively we have $S_l = \{(u_1, \ldots, u_k) \in T^k \mid \mathcal{T}, u_1, \ldots, u_k \models \alpha_l\}$ for each $l \leq n$, then as before we put $\mathcal{T}, t_1, \ldots, t_k \models \#(\alpha_1, \ldots, \alpha_n)$ iff $\mathcal{T} \models \tau_{\#}(t_1, \ldots, t_k, S_1, \ldots,$ $S_n)$.

For some examples of this style of logic, see, e.g., [64, chapter 7].

4. *Interval temporal logics.*

There are two different approaches to interval temporal logics.

In the approach of Moszkowski [138] and the Duration calculus [221] the propositional version of the interval temporal logic is defined on point-based linear temporal structures. Atoms are evaluated at points but we build up more complicated formulas which are evaluated on intervals. A variety of interval temporal logics can be defined by considering various linear temporal structures such as the integers, or a finite subset of them, or such as the reals (in Duration calculus). A variety of operators can be defined such as **empty**, $\bigcirc\alpha$ or $\alpha;\beta$. Consider, for example, a linear temporal structure $(\mathbb{N}, <, h)$. Formulas are evaluated at intervals $\sigma = \{t \in T \mid \sigma_- \leq t \leq \sigma_+\}$ of $(T, <)$: $(T, <, h), \sigma \models p$ iff $\sigma_- \in h(p)$; $(T, <, h), \sigma \models \bigcirc\alpha$ iff $(T, <, h), \sigma^+ \models \alpha$ where $\sigma = \{t \in T \mid \sigma_- + 1 \leq t \leq \sigma_+\}$; and $(T, <, h), \sigma \models \alpha;\beta$ iff there is some $z \in \mathbb{N}$ such that $(T, <, h), \{t \in T \mid \sigma_- \leq t \leq z\} \models \alpha$ and $(T, <, h), \{t \in T \mid z \leq t \leq \sigma_+\} \models \beta$. The Duration Calculus also allows some metric information to be specified, such as the length of intervals, and, via the duration operator, the integrated duration of the truth of a proposition during an interval. See [220] for more details on the Duration Calculus.

In the other approach, such as, for example, the interval temporal logics of Halpern and Shoham [87] and Venema [197], intervals themselves are the basic temporal units of structure and atoms are evaluated on intervals. Generally such logics have unary (or 1-place) modal operators corresponding to binary relations between intervals. We have seen in Section 2.6 above that there are 13 standard relations identified between intervals such as the i *precedes* j (i.e. $i < j$) relation. Thus, in [192], we have a modal diamond P (for precedes) with semantics:

$$\mathcal{I}, i \models Pp \text{ iff } \text{ there is an interval } j \text{ s.t. } j < i \text{ and } \mathcal{I}, j \models p.$$

In [197] we see a more powerful interval logic CDT extending this approach with some binary modal connectives inspired by natural language, computation [172]

and relation algebra. The *chop* connective C from this system (corresponding to ; of the Duration Calculus) is defined in terms of the accessibility relation $A(i,j,k)$ true when the following all hold: i and k begin together, i ends where j starts, and j and k end together. The semantics of chop is:

$$\mathcal{I}, k \models pCq \text{ iff there are intervals } i, j \text{ s.t. } A(i,j,k),\ \mathcal{I}, i \models p \text{ and } \mathcal{I}, j \models q.$$

There is a fuller treatment of interval temporal logics in [193]. See also [77].

5. *Combinations of temporal logic and other modal logics.*

There has been much recent interest in combinations of temporal logic and other modal logics. These include temporal-epistemic [48] and spatio-temporal [65] logics as well as logics of parallel time [120]. The simplest appropriate structure (from [189]) is a set W of classical propositional worlds ($h(p) \subseteq W$) endowed with both a temporal ordering $w_1 < w_2$ and an accessibility relation R_b for each box $\Box_b$ of the modal language. A combined language can have formulas including the atoms, and closed under forming $G\alpha$, $H\alpha$ and $\Box_b\alpha$ from any formula α. Letting $\mathcal{M} = (W, <, \{R_b\}, h)$, the semantic clauses are just:

$\mathcal{M}, w \models G\alpha$ iff for all $w' \in W$, if $w < w'$ then $\mathcal{M}, w' \models \alpha$;
$\mathcal{M}, w \models H\alpha$ iff for all $w' \in W$, if $w' < w$ then $\mathcal{M}, w' \models \alpha$;
$\mathcal{M}, w' \models \Box_b\alpha$ iff for all $w' \in W$, if wR_bw' then $\mathcal{M}, w' \models \alpha$.

One common way of constructing such a structure is as the cartesian product of a temporal structure and a modal structure. The reader is referred to Chapter 15 or [65] for details.

The 2-dimensional temporal logics of [130] and [49] are similar: we just have to distinguish temporal connectives operating in one dimension from those of another dimension.

Predicate temporal logic is a combination (of temporal logic with first-order logic) of a different nature. The standard predicate temporal logic with F and P of constant domains, rigid terms and rigid variable assignments is defined as follows.

Recall from Section 2.7 that the atomic formulas of our logic will be classical atomic first-order formulas from a signature L, consisting of relation symbols, function symbols, and constants. Assume that V is our set of (domain) variable symbols, i.e. those standing for elements of the object domain. The set of formulas will include all the atomic formulas and also $\neg\alpha$, $\alpha \wedge \beta$, $\forall x\alpha$, $F\alpha$ and $P\alpha$ for any formulas α and β and any domain variable symbol x.

A temporal structure will have the form $\mathcal{M} = (T, <, D, (M_t : t \in T))$, where $(T, <)$ is a flow of time, D is the domain of $\mathcal{M}$, and for each $t \in T$, M_t is an L-structure with domain D. An assignment to the variable symbols is just a map from V to D (with no dependence on T).

The assignment ν combines with the interpretations for the predicate and function symbols at time a to give us the interpretation $\nu_a(t) \in D$ for any term t at time a. In particular, $\nu_a(x) = \nu(x)$ for $x \in V$, $\nu_a(f(t_1, \ldots, t_n)) = f^{M_a}(\nu_a(t_1), \ldots, \nu_a(t_n))$.

For an assignment ν, we define truth of formulas at times in $\mathcal{M}$ in a straightforward way. The more interesting clauses are:

$\mathcal{M}, \nu, a \models p(t_1, \ldots, t_n)$ iff $(\nu_a(t_1), \ldots, \nu_a(t_n)) \in p^{M_a}$;
$\mathcal{M}, \nu, a \models \forall x \alpha$ iff for all $d \in D$, $\mathcal{M}, \nu[x \rightsquigarrow d], a \models \alpha$;
$\mathcal{M}, \nu, a \models F\alpha$ iff there is $b \in T$ such that $a < b$ and $\mathcal{M}, \nu, b \models \alpha$.
Here $\nu[x \rightsquigarrow d]$ is just the assignment which is the same as ν except that x is mapped to d.

It is worth mentioning the variation, called TLA or the *Temporal Logic of Actions* [121] on a predicate temporal logic over natural numbers time. This has received much attention in connection with reasoning about specifications of programs. The set of allowable formulas is restricted so that they are each *stuttering invariant*. This means that these formulas can not distinguish between behaviours (of a program) which only differ by having the same state (i.e. values of externally observable program variables) repeated consecutively a different number of times.

3.6 Temporal logic with second-order operations

All the temporal logics so far considered involve connectives with essentially first-order definitions. As we will soon see, each formula of such a logic can be translated into first-order logic in a meaning-preserving way. Sometimes, however, first-order expressivity is insufficient. For example, the property that an atom is true at every even number in a temporal structure $(\mathbb{N}, <, h)$ is not expressible with first-order connectives. We have already mentioned monadic second-order logic in the context of time. Now it is the turn of modal-style temporal logics to benefit from second-order operations.

As usual, there are several ways of introducing second-order devices into temporal logic. We consider four options in turn. The first two are generally used over natural numbers time $(\mathbb{N}, <)$, and finite linear flows; the second two can be used with any flow of time.

1. *Regular expressions.* The addition of these to propositional temporal logic was proposed by Wolper [205, 204, 206]. It can be done as follows. A (right-linear) *grammar* is a triple

$$\mathcal{G} = ((a^1, \ldots, a^k), Q, \delta),$$

where $k \geq 1$ is finite, $a^1, \ldots, a^k$ are distinct objects (the *terminals* of $\mathcal{G}$), Q is a finite non-empty set (the *non-terminals* of $\mathcal{G}$), and δ is a finite set of *rules* of the form $u \to av$ and $u \to a$, where $u, v \in Q$ and $a \in \{a^1, \ldots, a^k\}$. We write A for the set $\{a^1, \ldots, a^k\}$; the order of enumeration of A given in $\mathcal{G}$ will be significant below.

Given $\mathcal{G}$ as above, and $v_0 \in Q$, we define sets S_n $(n < \omega)$ and $S(\mathcal{G}, v_0)$. Intuitively, S_n is the set of words over $A \cup Q$ constructible from v_0 by n applications of rules in δ; $S(\mathcal{G}, v_0)$ is the set of finite or infinite words over A constructible from v_0 by $\leq \omega$ applications of the rules. The sets are defined formally by induction as follows. We let $S_0 = \{v_0\}$. Given S_n, we let $S_{n+1} = \{a_1 \ldots a_{n+1}v : \exists u \in Q(a_1 \ldots a_n u \in S_n \wedge u \to a_{n+1}v \in \delta)\}$. Finally, we let $S(\mathcal{G}, v_0) = (A^* \cap \bigcup_{n<\omega} S_n) \cup \{a_1 a_2 \ldots : (\forall n < \omega)(\exists v \in Q) a_1 \ldots a_n v \in S_n\}$. Here A^* denotes the set of finite words over A; the effect of taking the intersection with it is to restrict to words without non-terminals.

As an example, if $\mathcal{G} = ((a,b), \{v\}, \{v \to av,\ v \to b\})$, then

$$\begin{aligned} S_0 &= \{v\}, \\ S_n &= \{\overbrace{aa\ldots a}^{n}v,\ \overbrace{aa\ldots a}^{n-1}b\}, \text{ for } 0 < n < \omega, \\ S(\mathcal{G}, v) &= \{b, ab, aab, aaab, \ldots\} \cup \{aaaaa\ldots\}. \end{aligned}$$

We remark that a grammar can be viewed as a finite-state automaton, with alphabet A, state set Q, and transition table δ. It follows that the sets of the form 'the set of all finite words in $S(\mathcal{G}, v)$', for arbitrary $\mathcal{G}$ and non-terminal v of $\mathcal{G}$, are precisely the regular languages.

Now let $\mathcal{T}$ be any propositional temporal logic suitable for temporal structures $\mathcal{M} = (\mathbb{N}, <, h)$ based on natural numbers time, where $h : L \to \wp(\mathbb{N})$. The extended temporal logic ETL($\mathcal{T}$) is then defined as follows. Its atomic formulas are $\top$, $\bot$, and q for each $q \in L$. More complex formulas are formed in two ways. First, given any n-ary connective $\sharp$ of $\mathcal{T}$, and ETL($\mathcal{T}$)-formulas $\alpha_1, \ldots, \alpha_n$, we can form the ETL($\mathcal{T}$)-formula $\sharp(\alpha_1, \ldots, \alpha_n)$; it is evaluated in $\mathcal{M}$ in the usual way. Second, given any grammar $\mathcal{G} = ((a^1, \ldots, a^k), Q, \delta)$, any $v \in Q$, and ETL($\mathcal{T}$)-formulas α_1, $\ldots, \alpha_k$, we can form the ETL($\mathcal{T}$)-formula $\mathcal{G}_v(\alpha_1, \ldots, \alpha_k)$. Then $\mathcal{M}, t \models \mathcal{G}_v(\alpha_1, \ldots, \alpha_k)$ iff there is a word $a^{i_0}a^{i_1}\ldots$ in $S(\mathcal{G}, v)$ of length $l \le \omega$, say, where $1 \le i_j \le k$ for each $j < l$, such that for all $j < l$ we have $\mathcal{M}, t + j \models \alpha_{i_j}$.

For example, if we take the grammar $\mathcal{G} = ((a,b), \{v\}, \{v \to av,\ v \to b\})$ mentioned above, then $\mathcal{M}, t \models \mathcal{G}_v(\alpha, \beta)$ iff either there is $u \ge t$ such that $\mathcal{M}, u \models \beta$ and $\mathcal{M}, v \models \alpha$ for all v with $t \le v < u$, or else $\mathcal{M}, u \models \alpha$ for all $u \ge t$. Roughly, because the words in $S(\mathcal{G}, v)$ are of the form $aaa\ldots ab$ or $aaa\ldots$, and we associate α with a and β with b, we require the pattern $\alpha\alpha\alpha\ldots\alpha\beta$ or $\alpha\alpha\alpha\ldots$ to begin at t. This is a weak form of the 'non-strict until' considered earlier. It is easily checked that $\mathcal{G}_v(\alpha, \bot)$ expresses a non-strict form of $G\alpha$, and $\neg\mathcal{G}_v(\neg\alpha, \bot)$ expresses the non-strict eventuality $F_{\le}\alpha$. So $\mathcal{G}_v(\alpha, \beta) \wedge \neg\mathcal{G}_v(\neg\beta, \bot)$ expresses the non-strict $\alpha U_{\le}\beta$ seen in Section 3.4.

In fact, all 'future-oriented' connectives examined earlier can be expressed in extended temporal logic. Moreover, the logic is decidable over natural numbers time, with PSPACE-complete validity problem [206].

2. *Gabbay's fixed point operator [61, 64].* This can be added to the propositional temporal syntax with connectives U (Until) and Y (Yesterday), creating a system known as UYF. Its syntax is as follows. The atomic formulas are $\top$, $\bot$, and q for $q \in L$, and if α, β are UYF-formulas then so are $\neg\alpha$, $\alpha \wedge \beta$, $Y\alpha$, $U(\alpha, \beta)$, and $\varphi q\alpha$, for any atom $q \in L$ such that α is *pure past in q* — this means that every occurrence of q in α that is not in the scope of a φq is in the scope of a Y and not in the scope of an U.

   The semantics of $\wedge, \neg, Y, U$ are as before. The semantics of φ is a little harder to define. Suppose that the formula α is pure past in q, and (inductively) that its semantics has been defined. In any temporal structure $(\mathbb{N}, <, h)$, it turns out (by examining the inductively-defined semantics of α) that the truth value of α at any time $t \in \mathbb{N}$ only depends on the values of q at times $0, 1, \ldots, t-1$. This means that we may define sets $S_t^h \subseteq \{0, 1, \ldots, t\}$ (for $t \in \mathbb{N}$) recursively as follows:

(a) $0 \in S_0^h$ iff $(\mathbb{N}, <, h), 0 \models \alpha$,

(b) for $t > 0$, if $u \leq t$ then $u \in S_t^h$ iff $(\mathbb{N}, <, h_t), u \models \alpha$, where h_t is the same as h except that $h_t(q) = S_{t-1}^h$.

Because α is pure past in q, if $0 \leq t \leq u$ then $t \in S_t^h$ iff $t \in S_u^h$. This means that the S_t^h 'converge' to a value $S^h = \{t \in \mathbb{N} : t \in S_t^h\}$. We then let S^h be the set of times at which $\varphi q \alpha$ is true: we define $(\mathbb{N}, <, h), t \models \varphi q \alpha$ iff $t \in S^h$.

Intuitively, what is going on is that we evaluate α at time 0: its value is independent of $h(q)$ since α is pure past in q. Then, we evaluate α at time 1; now the value of q at 0 does matter (though its values at $1, 2, \ldots$ do not), and we let it be the same value that we obtained for α at 0. Then we evaluate α at 2, giving q the same values at $0, 1$ as α had; and so on. For example, $\varphi q \neg Y q$ evaluates to true at 0, then false at 1, true at 2, and so on: it is true at precisely the even times. We can also express $S(\alpha, \beta)$, by $\varphi q Y(\alpha \vee (q \wedge \beta))$.

It turns out that $\varphi q \alpha$ defines a *fixed point* of the operator $\Phi^h : \wp(\mathbb{N}) \to \wp(\mathbb{N})$ given by

$$\Phi^h : S \mapsto \{t \in \mathbb{N} : (\mathbb{N}, <, h_{q \rightsquigarrow S}), t \models \alpha\},$$

$$\text{where} \quad h_{q \rightsquigarrow S}(p) = \begin{cases} h(p), & \text{if } p \in L \setminus \{q\}, \\ S, & \text{if } p = q. \end{cases}$$

This operator is not monotonic, but because of the pure past restriction, it does in fact have a unique fixed point. However, the 'recursive' view of φ is also natural.

The φ operator was introduced in [61], via a logic called USF expressively equivalent to the UYF just defined. [95] and [64, chapter 8] showed that any UYF-formula is equivalent to one with at most two nested φs, and that UYF is decidable with PSPACE-complete validity problem. They also prove that UYF is expressively equivalent to monadic second-order logic over $(\mathbb{N}, <)$ (see Section 3.2). For any monadic second-order formula $\psi(t, Q_1, \ldots, Q_n)$, where $Q_1, \ldots, Q_n$ are (free) unary relation symbols, there is a UYF-formula $\alpha(q_1, \ldots, q_n)$ such that for any $h : L \to \wp(\mathbb{N})$, and any $t \in \mathbb{N}$, we have $(\mathbb{N}, <, h), t \models \alpha$ iff $(\mathbb{N}, <) \models \psi(t, h(q_1), \ldots, h(q_n))$. This is a second-order form of *expressive completeness,* the first-order form of which will be discussed in Section 4.

3. Moving to arbitrary flows of time, we can add the *μ-calculus* to basic temporal logic. Over $(\mathbb{N}, <)$, this provides expressive power equal to monadic second-order logic; but its use is not limited to this flow. We will not consider it in detail here, because chapter 12 is devoted to it.

4. Another option is to include *second-order quantification.* In its simplest form this means adding, to propositional temporal logic, an ability to quantify over propositional atoms. Given a temporal formula α and an atom q, we may admit the formula $\exists q \alpha$ to our logic. In such a formula, q is a 'bound variable'. The $\exists q$ can be interpreted in several ways:

   - $(T, <, h), t \models_a \exists q \alpha$ iff $(T, <, h) \models_a \alpha(r/q)$ for some $r \in L$; here, $\alpha(r/q)$ denotes the result of replacing all free occurrences of q by r throughout α. This semantics reads $\exists q \alpha$ as $\bigvee_{r \in L} \alpha(r/q)$. We quantify over existing or actual values of atoms (hence the 'a').

- $(T, <, h), t \models_2 \exists q\alpha$ iff $(T, <, g) \models_2 \alpha$ for some assignment $g : L \to \wp(T)$ agreeing with h except perhaps on q (that is, $g(r) = h(r)$ for all $r \in L \setminus \{q\}$). This is straightforward monadic second-order quantification (hence the '2').
- More generally, we can specify a set $\mathcal{S} \subseteq \wp(T)$ over which $\exists$ ranges. We then stipulate that $(T, <, h), t \models_{\mathcal{S}} \exists q\alpha$ iff $(T, <, g) \models_{\mathcal{S}} \alpha$ for some assignment $g : L \to \wp(T)$ agreeing with h except perhaps on q and with $g(q) \in \mathcal{S}$. Setting $\mathcal{S} = \text{rng}(h)$ yields $\models_a$, and $\mathcal{S} = \wp(T)$ yields $\models_2$.

These systems are very powerful but computationally rather complex. In [64, chapter 8], an axiomatisation was given for $\models_a$, and $\models_a$ and $\models_2$ were both shown undecidable.

3.7 Branching Time Operations

Most work on branching time temporal logics, by which we mean those being evaluated on tree structures, divides into two schools. The oldest is the philosophical school studying logics of historical necessity in which arbitrary trees are allowed as flows of time. The other is the computing school studying logics of paths through transition systems which we have seen is closely related to logics of temporal structures with flows which are discrete ω-height trees. There are many common issues across the schools but the notation is different so we tackle each in turn.

Historical Necessity

In the philosophical tradition, temporal logics of branching time are often referred to as logics of historical necessity: there are no alternative past histories before any point of time. We present here a standard approach called Ockhamist logic with local assignment to atoms.

Recall that a branch (or *history*) of $(T, <)$ is a maximal linearly $<$-ordered subset of $(T, <)$. Let $\mathbf{B}(T, <)$ be the set of all branches of $(T, <)$.

Fix a countable set L of atoms. *Structures* $\mathcal{T} = (T, <, h)$ will have a tree frame $(T, <)$ and a valuation h for the atoms i.e. for each atom $p \in L$, $h(p) \subseteq T$.

The language HN is generated by the connectives G, H and $\Box$ along with classical $\neg$ and $\wedge$. That is, we define the set of formulas recursively to contain the atoms and for formulas α and β we include $\neg\alpha$, $\alpha \wedge \beta$, $G\alpha$, $H\alpha$ and $\Box\alpha$.

Formulas are evaluated at points on branches in structures. We write $\mathcal{T}, \sigma, x \models \alpha$ when α is true at the point x of the branch σ of the structure $\mathcal{T}$. This is defined recursively as follows. Suppose that we have defined the truth of formulas α and β at all points of all branches of $\mathcal{T}$. Then for all branches σ, for all points x of σ:

$\mathcal{T}, \sigma, x \models p$	iff	$x \in h(p)$, for p atomic;
$\mathcal{T}, \sigma, x \models \neg\alpha$	iff	$\mathcal{T}, \sigma, x \not\models \alpha$;
$\mathcal{T}, \sigma, x \models \alpha \wedge \beta$	iff	both $\mathcal{T}, \sigma, x \models \alpha$ and $\mathcal{T}, \sigma, x \models \beta$;
$\mathcal{T}, \sigma, x \models G\alpha$	iff	for all $y > x$ in σ we have $\mathcal{T}, \sigma, y \models \alpha$;
$\mathcal{T}, \sigma, x \models H\alpha$	iff	for all $y < x$ in σ we have $\mathcal{T}, \sigma, y \models \alpha$;
$\mathcal{T}, \sigma, x \models \Box\alpha$	iff	for every branch π containing x we have $\mathcal{T}, \pi, x \models \alpha$.

As well as the linear time abbreviations we also have $\Diamond\alpha \equiv \neg\Box\neg\alpha$.

We say that α is *valid* in HN iff for all structures $\mathcal{T}$, for all branches σ in $\mathcal{T}$, for all points $x \in \sigma$, we have $\mathcal{T}, \sigma, x \models \alpha$. Let us write $\models \alpha$ in that case.

Variations and extensions on this traditional set-up abound. They include the addition of (strict) until and since operators [217], allowing the truth of propositions to be dependent on branch as well as time point (discussed by Prior in [150] and referred to as non-local assignment in [218] and [165]) and the *bundled* logics of [28] in which the modal quantification over branches is restricted to a given set.

An important class of branching time temporal logics are those termed Peircean branching time logics by Prior. These logics are defined so that truth of all formulas depends only on a time point of evaluation and not on a branch of evaluation. An example includes the sublanguage of the Ockhamist historical necessity logic above in which the branch and temporal modalities are only allowed in the combinations $\Box G$, $\Box F$, $\Box H$, $\Box P$, $\Diamond G$, $\Diamond F$, $\Diamond H$ and $\Diamond P$. See [150, 151, 216] for details.

PCTL, CTL*, CTL and QCTL*

A simple branching time temporal logic for computing applications called CTL for computational tree language was described in [35]. The branching is used to capture indeterminacy, choice or openness to the environment. A much more expressive language, CTL*, also called the full computational tree logic, was provided in [46] and [43] to extend, in expressiveness, both CTL and the linear PTL. We will work with a further extension here. The branching time logic PCTL* [123], [162] extends both CTL* and linear time temporal logic augmented with past-time operators.

The formulas of PCTL* are built from the atomic propositions in L recursively using classical connectives $\neg$ and $\wedge$ as well as the temporal connectives X, S, Y and U and the path quantifier A: if α and β are formulas then so are $X\alpha$, $\alpha S\beta$, $Y\alpha$, $\alpha U\beta$ and $A\alpha$.

Formulas are evaluated in *transition systems* (see definition 11 above). Since ω-height trees are transition systems this is a generalization of working with a tree shaped model of time.

Truth of formulas is evaluated at indexes in fullpaths in transition systems. We write $M, b, i \models \alpha$ iff the formula α is true at the index (time) i of the fullpath b in the transition system $M = (S, R, g)$. This is defined recursively by:

$$
\begin{array}{lll}
M, b, i \models p & \text{iff} & p \in g(b_i), \text{ any } p \in L \\
M, b, i \models \neg\alpha & \text{iff} & M, b, i \not\models \alpha \\
M, b, i \models \alpha \wedge \beta & \text{iff} & M, b, i \models \alpha \text{ and } M, b, i \models \beta \\
M, b, i \models X\alpha & \text{iff} & M, b, i+1 \models \alpha \\
M, b, i \models \alpha U\beta & \text{iff} & \text{there is some } j \geq i \text{ such that } M, b, j \models \beta \\
 & & \text{and for each } k, \text{ if } i \leq k < j \text{ then } M, b, k \models \alpha \\
M, b, i \models Y\alpha & \text{iff} & i > 0 \text{ and } M, b, i-1 \models \alpha \\
M, b, i \models \alpha S\beta & \text{iff} & \text{there is some } j \leq i \text{ such that } M, b, j \models \beta \\
 & & \text{and for each } k, \text{ if } j < k \leq i \text{ then } M, b, k \models \alpha \\
M, b, i \models A\alpha & \text{iff} & M, b', i \models \alpha \text{ for every fullpath } b' \text{ such that } b_{\leq i} = b'_{\leq i}
\end{array}
$$

We say that α is *valid* in PCTL*, and write $\models_P \alpha$, iff for all transition systems M, for all fullpaths b in M, for all indexes i, we have $M, b, i \models \alpha$.

We use the usual past-time linear temporal logic abbreviations plus $E\alpha \equiv \neg A \neg \alpha$.

The formulas of CTL* are just those of PCTL* which do not contain the past operators S or Y. The past before the index i of evaluation is irrelevant for CTL* formulas and so the semantics of CTL* on fullpaths through Kripke structures is usually presented with

no mention of such an index: truth is evaluated from the point of view of the beginning of the fullpath. The semantics of CTL* is defined so that $M, b \models A\alpha$ iff $M, b' \models \alpha$ for any fullpath b' through M starting at the same state as b does.

Note that formulas of CTL* which are boolean combinations of atoms and formulas of the form $A\gamma$ are often called *state formulas.* Their truth depends only on the point of evaluation independent of any fullpath. This is not true in the PCTL* semantics but it is easy to see that for such an α we do have $\models \alpha \leftrightarrow A\alpha$.

The original CTL language actually only consisted of some of the state formulas of CTL*. In fact it contains just boolean combinations of the atoms, and for formulas α and β of CTL, the formulas $EX\alpha$, $E(\alpha U\beta)$, $AX\alpha$ and $A(\alpha U\beta)$. CTL is a Peircean branching time logic.

There also exist branching time temporal logics which extend CTL* via propositional quantification (see for example [57]). The reader should also see Chapter 12 for a full coverage of the μ-calculus which is a popular extension of CTL*.

4 EXPRESSIVE POWER OF INTERNAL AND EXTERNAL PARADIGMS

In the preceding section, we introduced both 'internal' (modal style) and 'external' (first-order style) logics for handling time. Now we wish to compare the two approaches. We will consider their relative expressive power, over various kinds of flow of time, and the complexity of deciding validity for them. Such comparisons probably began with Kamp [112], and they have given rise to a rather rich and interesting field of work. Mainly we will restrict our attention to linear flows of time, but we will mention some results for trees and for arbitrary flows.

The most basic observation is that there is a 'standard' translation of propositional temporal formulas into first-order logic, and we will start with this in Section 4.1. The question then arises of how much of first-order logic is captured by this translation. Section 4.2 introduces the crucial notion of expressive completeness, whereby all first-order properties are expressible by temporal formulas. Expressive completeness is studied in Section 4.3 over all flows of time, and in Section 4.4 over linear flows, where we mention Kamp's famous theorem that Until and Since are expressively complete over Dedekind-complete linear time. Section 4.5 briefly discusses algorithmic issues. In Section 4.6 we discuss separation, an important notion due to Gabbay.

All this is for one-dimensional propositional temporal logic. In the last two sections, we broaden our view. In Section 4.7, we briefly consider expressive completeness for many-dimensional connectives. In recent times, expressive completeness for first-order temporal logic has also received attention, and we end in Section 4.8 by discussing some of this work.

Notation. We will reserve the phrase 'temporal logic' for a modal-style logic. We will use 'first-order style logic' or 'first-order logic' for the other kind. We will generally write temporal formulas as $\alpha, \beta, \ldots$, and classical first-order ones as φ, ψ, etc.

Until the end of Section 4.6, we will restrict our attention to the (one-dimensional) propositional case. Recall that L is our set of atoms, and that $L^\star$ denotes the first-order signature $\{<\} \cup \{Q : q \in L\}$, where the Q are unary relation symbols. For a temporal structure $\mathcal{M} = (T, <, h)$, $\mathcal{M}^\star$ is the $L^\star$-structure obtained from $\mathcal{M}$ in the natural way: see Section 3.1. We will write $\mathcal{L}$ for the class of all linear flows of time, and $\mathcal{D}$ for the

class of all Dedekind-complete linear flows (see definition 4).

4.1 Translating temporal logic into first-order logic

To begin, let us observe that the first-order approach subsumes any temporal logic whose temporal connectives have first-order definitions ('tables').

Suppose we have a set $\mathcal{T}$ of connectives, each $\sharp \in \mathcal{T}$ being defined by a first-order table $\tau_\sharp(t, P_1, \ldots, P_n)$ written with variables in a fixed set $\mathcal{V}$, say. For each $v \in \mathcal{V}$, let $\tau_\sharp(v, P_1, \ldots, P_n)$ denote the result of applying to the variables of $\tau_\sharp(t, P_1, \ldots, P_n)$ a permutation $\pi_\sharp$ of $\mathcal{V}$ that takes t to v. Then we may recursively translate each $\mathcal{T}$-formula $\alpha(q_1, \ldots, q_n)$ into a first-order $L^\star$-formula $\alpha^v(v, Q_1, \ldots, Q_n)$, with free variable v and written with variables in $\mathcal{V}$. The translation is defined by induction, as follows:

1. For an atom q, and $v \in \mathcal{V}$, we set $q^v = Q(v)$.
2. We let $\top^v = \top$ and $\bot^v = \bot$.
3. We define $(\neg\alpha)^v = \neg(\alpha^v)$, and $(\alpha \wedge \beta)^v = \alpha^v \wedge \beta^v$.
4. For each $\sharp \in \mathcal{T}$, we define $(\sharp(\alpha_1, \ldots, \alpha_n))^v$ to be the formula obtained from $\tau_\sharp(v, P_1, \ldots, P_n)$ by simultaneously replacing each atomic subformula $P_i(u)$ of it (where $i \leq n$ and $u \in \mathcal{V}$) by α_i^u.

As an example, consider the set $\mathcal{T}$ of connectives F, U, with tables as defined before, written with variables in $\mathcal{V} = \{t, u, v\}$. Then we could let

$$\begin{aligned}
(Fq)^t &= \exists u(u > t \wedge Q(u)),\\
U(p,q)^u &= \exists v(v > u \wedge Q(v) \wedge \forall t(u < t < v \to P(t))),\\
F\neg U(p,q)^t &= \exists u(u > t \wedge (\neg U(p,q))^u)\\
&= \exists u(u > t \wedge \neg\exists v(v > u \wedge Q(v) \wedge \forall t(u < t < v \to P(t)))).
\end{aligned}$$

The particular $\mathcal{V}$ used is generally unimportant, so long as it is large enough; so we will generally abuse notation slightly by writing $\alpha^\star(t, P_1, \ldots, P_n)$ or just $\alpha^\star(t)$ for the $L^\star$-formula α^t above, for any variable t. $\alpha^\star$ will be called the *standard translation* of α.

It should be clear that the standard translation is meaning-preserving. Formally, for any temporal structure $\mathcal{M} = (T, <, h)$, any $\mathcal{T}$-formula α, and any $a \in T$, we have

$$\mathcal{M}, a \models \alpha \iff \mathcal{M}^\star \models \alpha^\star(a, h(q_1), \ldots, h(q_n)).$$

This shows that first-order logic is at least as expressive as $\mathcal{T}$. Observe that we only need a $\mathcal{V}$ large enough to write the tables of the connectives of $\mathcal{T}$. So for example, the two-variable fragment of first-order logic is enough to express every formula of the temporal logic with F and P, and three variables suffice for U and S.

4.2 Expressive completeness

Clearly, the formulas $\{\alpha^x : \alpha \text{ a } \mathcal{T}\text{-formula}, x \in \mathcal{V}\}$ form, in general, a proper fragment of even the $\mathcal{V}$-variable fragment of first-order logic. For example, the formula $\exists y(y \neq x)$ is not of the form α^x for any α of the temporal logic with connectives F and P. This suggests that temporal logic is weaker than first-order logic; but to see if this is really

true, we have to consider not just equality but *equivalence* of first-order formulas to the translations of temporal ones. Of course, 'equivalence' is relative to the underlying flow(s) of time.

DEFINITION 12. Let $\mathcal{C}$ be a class of flows of time.

1. We say that first-order $L^\star$-formulas φ, ψ are *equivalent over* $\mathcal{C}$ if for every temporal structure $\mathcal{M} = (T, <, h)$ with $(T, <) \in \mathcal{C}$, and every assignment ν of the free variables of φ and ψ to elements of T, we have $\mathcal{M}^\star, \nu \models \varphi \leftrightarrow \psi$. Note that φ, ψ need not have the same free variables. For example, $v = v$ is equivalent to $\top$ over any class $\mathcal{C}$.

2. We say that an $L^\star$-formula $\varphi(t)$ is *equivalent over* $\mathcal{C}$ to a temporal formula α, if $\varphi(t)$ and $\alpha^\star(t)$ are equivalent over $\mathcal{C}$.

3. We will use obvious contractions: two formulas are said to be *equivalent over linear time* if they are equivalent over the class of all linear flows of time, *equivalent over a flow of time* $(T, <)$ if they are equivalent over the class $\{(T, <)\}$, and so on.

Since we are especially curious about whether temporal logic can ever match the power of first-order logic, we make the following definition.

DEFINITION 13. Let $\mathcal{C}$ be a class of flows of time, and $\mathcal{T}$ a temporal logic. We say that $\mathcal{T}$ is *(propositionally) expressively complete over* $\mathcal{C}$ if for every first-order $L^\star$-formula φ with one free variable, there is a $\mathcal{T}$-formula α that is equivalent to φ over $\mathcal{C}$.

This is somewhat analogous to the boolean connectives $\wedge, \neg$ being able to express any boolean function with any finite number of arguments. Since all temporal logics with connectives defined by first-order tables can be expressed in first-order logic, an expressively complete temporal logic would be an attractive proposition: it would be able to express all first-order-definable connectives. (Of course, it may not be able to express connectives defined by second-order tables; but here, we are focusing on the comparison of temporal logic with first-order logic. Aspects of second-order expressive completeness were discussed in Section 3.6.) Note that any temporal logic that is expressively complete over $\mathcal{C}$ remains expressively complete over any subclass of $\mathcal{C}$. Also note that if we are willing to allow infinitely many temporal connectives, we can add one for each $L^\star$-formula, yielding a temporal logic that is trivially expressively complete over any class of flows of time. So we will concentrate on temporal logics with *finitely many connectives.* (Another possibility would be to impose a bound on the number of variables in the defining tables of connectives.) Some natural questions now arise (brief answers are in italics):

1. Is there a temporal logic that is expressively complete over the class of all flows of time? *No: see theorem 14.*

2. If not, then for which classes $\mathcal{C}$ of flows of time does there exist an expressively complete temporal logic? *Examples include the linear and Dedekind-complete linear flows: see, e.g., theorem 15. That $\mathcal{C}$ has finite 'Henkin dimension' is a necessary but not sufficient condition; see Section 4.7.*

3. How can we tell whether a given temporal logic is expressively complete (over a given class $\mathcal{C}$ of flows of time)? *Separation is useful: see theorem 17.*

4. If $\mathcal{T}$ is expressively complete over $\mathcal{C}$, is there an effective procedure to convert an arbitrary first-order formula $\varphi(x, P_1, \ldots, P_n)$ into a $\mathcal{T}$-formula α whose standard translation is equivalent to φ over $\mathcal{C}$? How efficient can such a procedure be? *Sometimes; usually non-elementary. See Section 4.5.*

5. If $\mathcal{T}$ is expressively complete over $\mathcal{C}$, what can we say about the minimal length of a $\mathcal{T}$-formula equivalent over $\mathcal{C}$ to a given first-order formula? This is asking how *succinct* $\mathcal{T}$ is, compared with first-order logic. *Sometimes there is a non-elementary blow-up. See Section 4.5.*

6. If $\mathcal{T}$ is not expressively complete over $\mathcal{C}$, is it decidable whether a given first-order formula is equivalent over $\mathcal{C}$ to the standard translation of some $\mathcal{T}$-formula? With what complexity? *Not known. See Section 4.5(4).*

Quite a lot of work has been done on questions like these. We now proceed to give some more detailed answers than above.

4.3 The class of all flows of time

We have seen that given a set $\mathcal{T}$ of temporal connectives whose tables are written using variables in a set $\mathcal{V}$, any $\mathcal{T}$-formula can be translated to a first-order formula written with variables in $\mathcal{V}$. For finite $\mathcal{T}$, a finite set $\mathcal{V}$ will always suffice. This means that there is a finite bound on the number of variables required to write the standard translation $\alpha^\star$ of any $\mathcal{T}$-formula α. As Gabbay showed in [59], it follows that over any class $\mathcal{C}$ of flows of time on which the expressive power of $L^\star$-formulas with one free variable increases infinitely often as the total number of available variables increases, there can be no expressively complete temporal logic with finitely many temporal connectives. This is so for the class of *all* flows of time — for example, for flows of the form $(T, \emptyset)$, it takes n variables to express that $|T| \geq n$. The answer to our first question is therefore 'no':

THEOREM 14 (Gabbay, [59]). *There is no temporal logic with finitely many connectives (with first-order tables) that is expressively complete over the class of all flows of time.*

In Section 4.7, we will say a little more about the connection of expressive completeness to numbers of variables.

4.4 Linear flows — Kamp's theorem

Quite remarkably, over *linear time* the picture is more positive. Recall that we let $\mathcal{L}$ denote the class of all linear flows of time. The negative argument sketched above for the class of all flows does not apply to $\mathcal{L}$: [148, 106] showed that every first-order $L^\star$-formula $\varphi(t, P_1, \ldots, P_n)$ can be equivalently rewritten over $\mathcal{L}$ using only three variables. This is not in itself sufficient to ensure the existence of an expressively complete temporal logic over it [93]. Nonetheless, Kamp showed that:

THEOREM 15 (Kamp, [112]). *The temporal logic with Until and Since is expressively complete over the class $\mathcal{D}$ of all Dedekind-complete linear flows of time.*

This seminal theorem initiated the whole study of expressive completeness, and it remains one of the most interesting and distinctive results in temporal logic; very few, if any, similar 'modal' results exist. Several alternative proofs of it and stronger results

have appeared; none of them are trivial (at least to most people). Kamp's proof is in [112]. A second proof uses *separation,* to be discussed in Section 4.6 below. A third line of proof, reminiscent of classical quantifier elimination, began in [66]. Expressive completeness of Until and Since over $(\mathbb{N}, <)$ was outlined, together with a statement that two further connectives can be added to Until and Since to create a temporal logic that is expressively complete over $\mathcal{L}$. This second statement was proved by Stavi in an unpublished manuscript; the extra connectives are now called the *Stavi connectives* (see Section 3.4). A game-based account of the method appeared in [63, 64]. The proof for Dedekind complete time is simpler and appeared in [94]; it was later streamlined further by Wilke (unpublished).

4.5 Computational issues

Even when we have an expressively complete temporal logic $\mathcal{T}$ over a class $\mathcal{C}$ of flows of time, still there are the issues of whether there is an effective procedure (algorithm) to obtain a $\mathcal{T}$-formula $\alpha_\varphi(p_1, \ldots, p_n)$ that is $\mathcal{C}$-equivalent to any given first-order formula $\varphi(x, P_1, \ldots, P_n)$, what the algorithm's complexity might be, and what might be the length of α_φ in terms of the length of φ (succinctness). We now make some observations on these questions.

1. If the universal monadic second-order theory of $\mathcal{C}$ is decidable — as, for example, it is for $\mathcal{L}$ and $\mathcal{D}$ [33] — by an algorithm $\mathcal{A}$, say, then there is an algorithm to obtain α_φ. For, given $\varphi(x, P_1, \ldots, P_n)$, we may enumerate all $\mathcal{T}$-formulas $\alpha(p_1, \ldots, p_n)$, check using $\mathcal{A}$ whether
$$\mathcal{C} \models \forall P_1 \ldots P_n \forall x(\alpha^*(x, P_1, \ldots, P_n) \leftrightarrow \varphi(x, P_1, \ldots, P_n)),$$
and print out α if yes. By expressive completeness, this process will terminate.

2. However, even over $\mathcal{C} = \{(\mathbb{N}, <)\}$, and with $\mathcal{T}$ the temporal connectives Until and Since, there is no elementary algorithm to obtain an equivalent $\mathcal{T}$-formula to every first-order formula. (An algorithm is *elementary* if it runs in time bounded by $2^{2^{\cdot^{\cdot^{2^n}}}}$ on all inputs of length n, for some stack of 2s of arbitrary but fixed height.) To see this, we note that the validity problem for $\mathcal{T}$ over $\mathcal{C}$ is PSPACE-complete [184]; in particular, there is a PSPACE algorithm $\mathcal{B}$ to decide it. If there were an elementary algorithm to translate first-order formulas into equivalent $\mathcal{T}$-formulas, we could combine it with $\mathcal{B}$, yielding an elementary decision procedure for the universal monadic second-order theory of $(\mathbb{N}, <)$. This contradicts the result of [134, p. 479], [186] that there is no elementary decision procedure for that logic.

3. This does not rule out the possibility that for every first-order φ there is a relatively short equivalent $\mathcal{T}$-formula α_φ, even if there is no elementary algorithm to construct one. But Etessami and Wilke showed in [47] that there is in general a non-elementary gap between the length of a first-order formula and the length of any equivalent temporal formula, over $(\mathbb{N}, <)$ and with respect to the temporal logic with Until and Since. Succinctness is currently a rather active area; see, e.g., [5, 82].

4. Until and Since are not expressively complete over all linear time. Rabinovich has asked if it is decidable whether an arbitrary first-order $L^\star$-formula $\varphi(t, P_1, \ldots, P_n)$

is equivalent over linear time to a temporal formula written with Until and Since. We are not aware of any answer to this question or any of its obvious variants.

4.6 Separation

Roughly, a temporal logic has the separation property if each of its formulas can be equivalently rewritten as a boolean combination of parts depending only on the past, present, and future. Surprisingly, separation is often equivalent to expressive completeness, and is an important method for proving expressive completeness.

We have seen temporal logics such as the logic with F and P, and that with U and S, suitable for linear time. It is natural to observe that formulas such as Pq and $S(q, \neg S(\neg q, r))$ depend only on the past; $q \to r$ depends only on the present and is not temporal at all; FGq and $U(q \wedge \neg U(\top, q), \neg q)$ depend only on the future; and $F(q \wedge Pr)$ has a mixed dependence on past, present, and future. Let us be more precise.

DEFINITION 16. (Gabbay, [61]) Let $\mathcal{C}$ be a class of flows of time. A formula α of a temporal logic $\mathcal{T}$ is said to be

- *pure past over $\mathcal{C}$,* if for any $(T, <) \in \mathcal{C}$, any $t \in T$, and any assignments $g, h : L \to \wp(T)$, if $u \in g(q) \iff u \in h(q)$ for all $u \in T$ with $u < t$ and all $q \in L$, then $(T, <, g), t \models \alpha$ iff $(T, <, h), t \models \alpha$;
- *pure present over $\mathcal{C}$,* for any $(T, <) \in \mathcal{C}$, any $t \in T$, and any assignments $g, h : L \to \wp(T)$, if $t \in g(q) \iff t \in h(q)$ for all $q \in L$, then $(T, <, g), t \models \alpha$ iff $(T, <, h), t \models \alpha$;
- *pure future over $\mathcal{C}$,* if for any $(T, <) \in \mathcal{C}$, any $t \in T$, and any assignments $g, h : L \to \wp(T)$ if $u \in g(q) \iff u \in h(q)$ for all $u \in T$ with $u > t$ and all $q \in L$, then $(T, <, g), t \models \alpha$ iff $(T, <, h), t \models \alpha$;
- *pure over $\mathcal{C}$,* if it is pure past, pure present, or pure future over $\mathcal{C}$,
- *separated over $\mathcal{C}$,* if it is a boolean combination of formulas that are pure over $\mathcal{C}$.

$\mathcal{T}$ is said to have the *separation property over $\mathcal{C}$* if every $\mathcal{T}$-formula is equivalent over $\mathcal{C}$ to a formula that is separated over $\mathcal{C}$.

So a formula is pure past if its truth value at any time depends only on the values of its atoms in the past. This definition is semantic, and couched in terms of atoms. This leads to some oddities that should perhaps be borne in mind. For example, over linear time, $F\top$ is pure present and pure past, as well as pure future; the formula $P\neg U(\top, \neg q)$ is actually pure past, even though it involves Until, and over dense flows like $(\mathbb{Q}, <)$ and $(\mathbb{R}, <)$, it cannot be equivalently rewritten using only Since. The dependence on the underlying flow of time can also lead to surprises. For instance, the formula $S(q, \bot)$ is pure future over $(\mathbb{Q}, <)$, since it is equivalent to $\bot$ over this flow; but it is not pure future over $(\mathbb{N}, <)$.

Now we can connect separation to expressive completeness. This is surprising because the two conditions seem unrelated.

THEOREM 17 (Gabbay; see [61] and [64, §9.3]). *Let $\mathcal{C}$ be a class of linear flows of time, and $\mathcal{T}$ a temporal logic able to express F and P over $\mathcal{C}$. Then $\mathcal{T}$ is expressively complete over $\mathcal{C}$ iff it has the separation property over $\mathcal{C}$.*

Proof. '$\Rightarrow$' is proved by showing that any first-order formula can be separated. The details are in [64].

For '$\Leftarrow$', assume that $\mathcal{T}$ has the separation property. We show that any first-order formula $\varphi(x, P_1, \ldots, P_n)$, with one free variable, x, and unary predicates $P_1, \ldots, P_n$, is equivalent over $\mathcal{C}$ to some $\mathcal{T}$-formula, by induction on the quantifier depth k of φ.

If k is 0, then φ is a boolean combination of formulas of the form $P_i(x)$, $x = x$, and $x < x$, which can be replaced by p_i, $\top$, and $\bot$, respectively, to obtain the result. Assume the result for k. It suffices to find a $\mathcal{T}$-formula equivalent over $\mathcal{C}$ to $\exists y\varphi(x, y, P_1, \ldots, P_n)$, where φ has quantifier depth k. We can suppose that x and y do not occur bound in φ.

Our strategy is first to remove the particular variable x from $\exists y\varphi$. The atomic subformulas of $\exists y\varphi$ involving x are of the form $P_i(x)$, $x = x$, $x < x$, $z = x$, $z < x$, and $z > x$, where z is some variable other than x. We can replace $x = x$ by $\top$ and $x < x$ by $\bot$. Next, we try to remove the occurrences of x in subformulas $P_i(x)$. For each $S \subseteq \{1, 2, \ldots, n\}$, let φ^S be the result of replacing each atomic subformula $P_i(x)$ of φ by $\top$ if $i \in S$, and by $\bot$ if $i \notin S$. φ^S still has the same quantifier depth, k, but has no occurrences of any $P_i(x)$. (It may of course involve $P_i(z)$ for variables z other than x.) Then $\exists y\varphi$ is equivalent to

$$\bigwedge_{S\subseteq\{1,\ldots,n\}} \Big(\bigwedge_{i\in S} P_i(x) \wedge \bigwedge_{j\notin S} \neg P_j(x) \rightarrow \exists y\varphi^S(x, y, P_1, \ldots, P_n)\Big).$$

The formulas $P_i(x)$ are equivalent to p_i, of course. So it is enough if we can express the formulas $\exists y\varphi^S(x, y, P_1, \ldots, P_n)$ as $\mathcal{T}$-formulas.

Let ψ be one of the formulas φ^S. All occurrences of x in atomic subformulas of ψ are of the form $x = z$, $x > z$, and $x < z$. To remove even these, temporarily introduce new atoms $r_=$, $r_<$, $r_>$, with corresponding unary predicates $R_=$, etc. Replace each atomic subformula $x = z$ in ψ by $R_=(z)$. Similarly, replace $z > x$ by $R_>(z)$, and $z < x$ by $R_<(z)$. We obtain a new formula $\psi'(y, P_1, \ldots, P_n, R_<, R_=, R_>)$, of quantifier depth k, in which x does not occur at all. *If we agree to interpret $R_=$ as $\{x\}$, $R_<$ as $\{t : t < x\}$, and $R_>$ as $\{t : t > x\}$, then $\exists y\psi(x, y, P_1, \ldots, P_n)$ is equivalent to $\exists y\psi'(y, P_1, \ldots, P_n, R_<, R_=, R_>)$.*

Now we know by the inductive hypothesis that $\psi'(y, P_1, \ldots, P_n, R_<, R_=, R_>)$ is equivalent over $\mathcal{C}$ to some $\mathcal{T}$-formula $\alpha(p_1, \ldots, p_n, r_<, r_=, r_>)$. So, as the flows in $\mathcal{C}$ are linear, $\exists y\psi'$ is equivalent over $\mathcal{C}$ to $\beta = \alpha \vee F\alpha \vee P\alpha$. Thus, *under our agreement,* the formula $\exists y\psi(x, y, P_1, \ldots, P_n)$ is equivalent over $\mathcal{C}$ to $\beta(p_1, \ldots, p_n, r_<, r_=, r_>)$.

Finally, we remove the r. *Separate β.* We obtain a boolean combination $\gamma(p_1, \ldots, p_n, r_<, r_=, r_>)$ of pure $\mathcal{T}$-formulas that is equivalent over $\mathcal{C}$ to β. Consider a pure past formula $\delta(p_1, \ldots, p_n, r_<, r_=, r_>)$ from this boolean combination. As δ is pure past, it only 'needs to know' the values of the atoms at points $t < x$ when we evaluate it at a point x. Its truth value is independent of their values at points $t \geq x$. So let us replace $r_=$ and $r_>$ in δ by $\bot$, as under our agreement, these atoms are equivalent to $\bot$ at all $t < x$. Replace $r_<$ by $\top$ — these are also equivalent before x. We obtain $\delta^*(p_1, \ldots, p_n) \stackrel{\text{def}}{=} \delta(p_1, \ldots, p_n, \top, \bot, \bot)$. Then *subject to our agreement,* the truth values of δ^* and δ at x are the same. We conduct a similar replacement campaign for each pure formula δ in γ. If δ is pure present, $r_=$ is replaced by $\top$, and the others by $\bot$. For pure future δ, $r_>$ is replaced by $\top$ instead. The result is a boolean combination $\gamma^*(p_1, \ldots, p_n)$, which, subject to our agreement, is equivalent to $\exists y\psi(x, y, P_1, \ldots, P_n)$.

But the agreement concerned atoms that do not appear in γ^* or ψ. So it is irrelevant. Thus, without any restriction on assignments to atoms, γ^* is equivalent over $\mathcal{C}$ to $\exists y\psi(x, y, P_1, \ldots, P_n)$. This completes the proof. ❑

THEOREM 18 (Gabbay; see [61] and [64, 10.2.9]). *The temporal logic with Until and Since has the separation property over* $(\mathbb{N}, <)$ *(i.e., over the class* $\{(\mathbb{N}, <)\}$*).*

THEOREM 19 (Gabbay, Reynolds; see [64, §10.3–11]). *The temporal logic with Until and Since has the separation property over* $(\mathbb{R}, <)$. *The temporal logic with Until, Since, and the Stavi connectives has the separation property over the class* $\mathcal{L}$ *of all linear flows of time.*

Hence, Until and Since are expressively complete over $(\mathbb{N}, <)$ and $(\mathbb{R}, <)$, and Until, Since, and the Stavi connectives are expressively complete over all linear flows. The proofs of theorems 18 and 19 are direct, showing that each formula can be separated. They are tough and tougher, respectively. Nonetheless, they are effective, and so, whilst not quite providing an algorithm to determine if a set of connectives is expressively complete, they do suggest a potential way of telling in practice whether a given set of connectives is expressively complete — in Gabbay's words, *try to separate and see where you get stuck!* The process may suggest additional connectives that are more nearly expressively complete. For example, in the temporal logic with F and P, an attempt to find a separated formula equivalent over linear time to $F(q \wedge Hr)$ shows the need for Until. This formula cannot be separated using only F and P, but it is equivalent to $Hr \wedge r \wedge U(r, q)$, which is separated over linear time.

Suppose we have a $\mathcal{T}$ able to express F and P and with the separation property over a class $\mathcal{C} \subseteq \mathcal{L}$. The proofs of the above theorems can provide an effective way of constructing, for any temporal formula α, a separated (over $\mathcal{C}$) $\mathcal{T}$-formula that is equivalent to α over $\mathcal{C}$. However, there are some outstanding open questions:

1. What is the optimal complexity of algorithms that, given a $\mathcal{T}$-formula α, output a separated formula equivalent to α over $\mathcal{C}$?

   The chief concrete instances of this problem are for U, S over $(\mathbb{N}, <)$ and over Dedekind complete time, and for U, S, and the Stavi connectives over linear time. It is known that all such algorithms require at least exponential time; it is likely that there is no elementary algorithm even for $(\mathbb{N}, <)$.

2. What can one say about the length of a shortest separated $\mathcal{T}$-formula equivalent (over $\mathcal{C}$) to a given $\mathcal{T}$-formula?

   Results in [122] imply that over $(\mathbb{N}, <)$, separation causes at least an exponential blow-up in length for some formulas.

We end this section by mentioning some separation and expressive completeness results for non-linear time. For ω-trees with a bounded number of immediate successors of each node, or similar such restrictions, see [10, 11, 12, 13, 106, 175]. For the language XPath over trees with a left-right ordering, see [128]. [136] shows that the expressive power of the branching time logic CTL* coincides with that of the class of bisimulation invariant properties expressible in so-called monadic path logic: monadic second order logic in which set quantification is restricted to paths. In order to prove this result, the authors first prove a composition theorem for trees. This approach is adapted from the proof in [86] that CTL* coincides with the whole of monadic path logic over the class of full binary trees.

4.7 Many-dimensional temporal logic

We can generalise the notion of standard translation (Section 4.1) in the obvious way to the many-dimensional temporal logics discussed in Section 3.5(3). For a set $\mathcal{T}$ of k-dimensional temporal connectives, the standard translation $\alpha^\star$ of a $\mathcal{T}$-formula α is an $L^\star$-formula with (in general) k free variables. For a class $\mathcal{C}$ of flows of time, $\mathcal{T}$ is said to be *(propositionally) expressively complete over* $\mathcal{C}$ if every $L^\star$-formula with k free variables is equivalent over $\mathcal{C}$ to (the standard translation of) a $\mathcal{T}$-formula. We briefly consider now what can be said about this notion. We will say more in the first-order case, in Section 4.8.

How does many-dimensional expressive completeness relate to the one-dimensional expressive completeness discussed above? Suppose we have a class $\mathcal{C}$ of flows of time. If $\mathcal{T}$ is a set of k-dimensional temporal connectives that is propositionally expressively complete over $\mathcal{C}$, we can very easily construct a finite set of $(k+1)$-dimensional connectives that are also expressively complete over $\mathcal{C}$. [64, theorem 13.3.4] has details of this 'dimension-boosting' technique.

On the other hand, there are classes of flows of time for which there is a finite expressively complete set of 2-dimensional temporal connectives but no finite expressively complete one-dimensional set. An example based on circular time is given in [64, theorem 13.7.12].

There are some partial characterisations of when many-dimensional expressive completeness can be expected. A class $\mathcal{C}$ of flows of time is said to have the *k-variable property* if every $L^\star$-formula φ with at most k free variables is equivalent over $\mathcal{C}$ to a formula written with k variables altogether. For example, the class $\mathcal{L}$ of linear flows has the 3-variable property [106, 148]. It was shown in [64, theorem 13.6.7] that if $\mathcal{C}$ has the k-variable property then there is a finite expressively complete set of k-dimensional temporal connectives for $\mathcal{C}$, and if $k \geq 3$, there is even a finite expressively complete set of $(k-1)$-dimensional connectives. (The k-variable property does not imply one-dimensional expressive completeness, because the circular-time example just mentioned has the 3-variable property.)

The related notion of 'Henkin dimension' was considered in [59, 93, 99]. A class $\mathcal{C}$ of flows of time has *Henkin dimension at most k* if every $L^\star$-formula can be equivalently rewritten over $\mathcal{C}$ to use at most k bound variables. In [64, theorem 13.2.4], it is shown that $\mathcal{C}$ having finite Henkin dimension is a necessary condition for there to exist a finite expressively complete set of finite-dimensional temporal connectives for $\mathcal{C}$. For a rather weaker notion of expressive completeness, it is necessary and sufficient.

Obtaining expressively complete connectives over a class of flows of time with the k-variable property or finite Henkin dimension is relatively straightforward. The connectives are many-dimensional and the dimensions can mimic the variables in first-order formulas, of which a bounded number are needed. So their expressive completeness may not be so surprising. Obtaining expressively complete *one-dimensional* connectives is a very different matter. Many-dimensional expressive completeness can be useful as a step on the road to proving expressive completeness for one-dimensional connectives (e.g., [175]), but one-dimensional results such as Kamp's theorem 15 are generally far more profound and difficult than many-dimensional ones, and have very different proofs.

So far, we have taken our temporal structures to have the form $(T, <, h)$, where $h : L \to \wp(T)$. In particular, the values of atoms depend on only a single time point. For k-dimensional temporal connectives, we can generalise our semantics and allow atoms to

depend on k time points: so $h : L \to \wp(T^k)$. This is what we called 'true' many-dimensional temporal logic in Section 3.5(3). If we do this, expressive completeness is no longer available. Venema showed in [198] that there is no finite set of 2-dimensional temporal connectives that is expressively complete over linear flows of time. His example uses a single atom, say q; for each $n \geq 1$, he defines a temporal structure $\mathcal{M}_n = (\mathbb{Q}, <, h_n)$ in which $h_n(q)$ is an equivalence relation with n classes, each of which is dense in $(\mathbb{Q}, <)$. We know that given any finite set $\mathcal{T}$ of 2-dimensional temporal connectives, there is finite n such that the standard translation of every $\mathcal{T}$-formula can be written with n variables. But an Ehrenfeucht–Fraïssé game argument shows that no n-variable first-order sentence can distinguish between $\mathcal{M}_n$ and $\mathcal{M}_{n+1}$. The argument generalises easily to higher dimensions.

4.8 First-order temporal logic

The study of expressive completeness in the setting of first-order temporal logic probably began with Kamp's [113], and the field is currently quite active. The picture is not as nice as in the propositional case: there are a few positive results, but also strong negative ones.

Fix a first-order relational signature L. (We assume for simplicity that L is relational, with no function symbols or constants; but our methods are quite general.) Recall from Section 2.7 that first-order temporal structures are of the form $\mathcal{M} = (T, <, D, (M_t : t \in T))$, where $(T, <)$ is a flow of time, and for some first-order signature L, each M_t is an L-structure with domain D.

We have seen two ways to describe these structures in logic. The first is to use a *temporal logic.* When we mention a set $\mathcal{T}$ of temporal connectives here, it will be implicit that the connectives in $\mathcal{T}$ all have the same finite dimension (perhaps greater than 1) and have first-order tables, but there is no implicit assumption that $\mathcal{T}$ is finite. Given a set $\mathcal{T}$ of connectives, we know how to form a first-order temporal logic $\mathcal{T}(L)$. For example, if $\mathcal{T}$ contains the unary one-dimensional connective F ('sometime in the future'), and L contains unary relation symbols dog and day, then

$$\alpha = \forall x(\mathsf{dog}(x) \to F\,\mathsf{day}(x))$$

('every dog will have his day') is a $\mathcal{T}(L)$-formula.

The second way is to use a *two-sorted first-order logic* over the $L^\star$-structure $\mathcal{M}^\star$ obtained from $\mathcal{M}$ as in Section 3.3. Recall that the two sorts are $\mathbf{t}$ (for time) and $\mathbf{d}$ (for the first-order domain). The domain of $\mathcal{M}^\star$ is the disjoint union of T (sort $\mathbf{t}$) and D (sort $\mathbf{d}$). $L^\star$ is obtained from L by adding an extra, $\mathbf{t}$-sorted coordinate to each relation symbol in L — so, for example, if $R \in L$ is an n-ary relation symbol, then we include in $L^\star$ a relation symbol $R^\star$ of sort $\mathbf{t} \times \mathbf{d}^n$. For $a \in T$ and $d_1, \ldots, d_n \in D$, we let $\mathcal{M}^\star \models R^\star(a, d_1, \ldots, d_n)$ iff $M_a \models R(d_1, \ldots, d_n)$. For $\mathcal{T}, L$ as above, an example of an $L^\star$-formula is

$$\varphi = \forall x(\mathsf{dog}^\star(t, x) \to \exists u(u > t \wedge \mathsf{day}^\star(u, x))).$$

We wish to compare the expressive power of these two approaches. To help distinguish them, we will write $\mathcal{T}(L)$-formulas as $\alpha, \beta, \ldots$, and $L^\star$-formulas as $\varphi, \psi, \ldots$. We write $\mathbf{d}$-variables as $x, y, z, x_1, \ldots, x_n$, etc, and $\mathbf{t}$-variables as $t, u, v, t_1, \ldots, t_m$, etc. We will use a, b, etc., for time elements, and d, e, etc., for domain elements.

Standard translation

Let us first observe that, analogously to the propositional case, each $\mathcal{T}(L)$-formula has a *standard translation* into two-sorted first-order logic. It can be defined as follows. Suppose that $\mathcal{T}$ consists of k-dimensional connectives $\sharp$ with first-order tables $\tau_\sharp(t_1,\ldots,t_k,P_1,\ldots,P_n)$, where $\sharp$ is n-ary and $P_1,\ldots,P_n$ are k-ary relation symbols . We regard all symbols in $\tau_\sharp$ as of sort $\mathbf{t}$. We can suppose that in each atomic subformula $P_i(u_1,\ldots,u_k)$ of $\tau_\sharp$, the variables $u_1,\ldots,u_k$ are distinct. As earlier, for any distinct $\mathbf{t}$-variables $u_1,\ldots,u_k$, $\tau_\sharp(u_1,\ldots,u_k,P_1,\ldots,P_n)$ denotes the result of permuting the variables in $\tau_\sharp(t_1,\ldots,t_k,P_1,\ldots,P_n)$ by a permutation taking $t_1,\ldots,t_k$ to $u_1,\ldots,u_k$, respectively. Then for any $\mathcal{T}(L)$-formula α and distinct $\mathbf{t}$-variables $t_1,\ldots,t_k$, we define the $L^\star$-formula $\alpha^{t_1,\ldots,t_k}$ by induction on α as follows:

$$\begin{aligned}
R(x_1,\ldots,x_m)^{t_1,\ldots,t_k} &= R^\star(t_1,x_1,\ldots,x_m), \text{ for each } m\text{-ary relation symbol } R\in L,\\
(x=y)^{t_1,\ldots,t_k} &= (x=y),\\
\top^{t_1,\ldots,t_k} &= \top, \text{ and similarly for } \bot,\\
(\alpha\wedge\beta)^{t_1,\ldots,t_k} &= \alpha^{t_1,\ldots,t_k}\wedge\beta^{t_1,\ldots,t_k},\\
(\neg\alpha)^{t_1,\ldots,t_k} &= \neg(\alpha^{t_1,\ldots,t_k}),\\
(\exists x\alpha)^{t_1,\ldots,t_k} &= \exists x(\alpha^{t_1,\ldots,t_k}).
\end{aligned}$$

Finally, for each n-ary connective $\sharp$ of $\mathcal{T}$, $\sharp(\alpha_1,\ldots,\alpha_n)^{t_1,\ldots,t_k}$ is defined to be the result of replacing each atomic subformula $P_i(u_1,\ldots,u_k)$ in $\tau_\sharp(t_1,\ldots,t_k,P_1,\ldots,P_n)$ by $\alpha_i^{u_1,\ldots,u_k}$. As before, we generally write $\alpha^{t_1,\ldots,t_k}$ simply as $\alpha^\star(t_1,\ldots,t_k,x_1,\ldots,x_m)$, and refer to $\alpha^\star$ as the *standard translation* of α. For example, the standard translation of α (above) is φ.

It should be clear that for any temporal structure $\mathcal{M}=(T,<,D,(M_t:t\in T))$, any $d_1,\ldots,d_m\in D$, any $a_1,\ldots,a_k\in T$, and any $\mathcal{T}$-formula $\alpha(x_1,\ldots,x_m)$, we have

$$\mathcal{M},a_1,\ldots,a_k\models\alpha(d_1,\ldots,d_m) \iff \mathcal{M}^\star\models\alpha^\star(a_1,\ldots,a_k,d_1,\ldots,d_m).$$

So the standard translation faithfully represents the meaning of the original temporal formula. It is therefore reasonable to generalise definition 12 to first-order temporal logic, as follows.

DEFINITION 20. Let $\mathcal{C}$ be a class of flows of time, and let $(T,<)$ be a flow of time.

1. We say that two-sorted first-order $L^\star$-formulas φ,ψ are *equivalent over* $\mathcal{C}$ if for every temporal structure $\mathcal{M}=(T,<,D,(M_t:t\in T))$ with $(T,<)\in\mathcal{C}$, and every sort-respecting assignment ν of the free variables of φ and ψ to elements of $T\cup D$, we have $\mathcal{M}^\star,\nu\models\varphi\leftrightarrow\psi$. Note as before that φ,ψ need not have the same free variables.

2. Let $\mathcal{T}$ be a set of k-dimensional temporal connectives, and let $(t_1,\ldots,t_k)$ be a sequence of distinct $\mathbf{t}$-variables. We say that a two-sorted $L^\star$-formula $\varphi(t_1,\ldots,t_k,x_1,\ldots,x_n)$ is *equivalent over* $\mathcal{C}$ to a first-order $\mathcal{T}(L)$-formula $\alpha(x_1,\ldots,x_n)$ if $\varphi(t_1,\ldots,t_k,x_1,\ldots,x_n)$ and $\alpha^\star(t_1,\ldots,t_k,x_1,\ldots,x_n)$, the standard translation of α, are equivalent over $\mathcal{C}$. (This definition implicitly depends on the choice of $t_1,\ldots,t_k$.)

3. For short, two formulas are said to be *equivalent over* $(T, <)$ if they are equivalent over the class $\{(T, <)\}$, *equivalent over linear time* if they are equivalent over the class of all linear flows of time, etc.

Standard translations have a restricted syntactic form. This will be important in the analysis to come. Formally:

DEFINITION 21. For an integer $k \geq 1$, let $L_k^\star$ be the fragment of $L^\star$ consisting of all formulas φ such that any subformula of φ of the form $\exists x \psi$, for any $\mathbf{d}$-variable x, has at most k free $\mathbf{t}$-variables.

LEMMA 22. *For any set $\mathcal{T}$ of k-dimensional temporal connectives, the standard translation of any $\mathcal{T}(L)$-formula is in $L_k^\star$.*

Proof. This is proved easily from the definitions by induction on $\mathcal{T}(L)$-formulas, and is essentially because $\mathcal{T}(L)$-formulas are evaluated at k time points. We leave the details to the reader. ❑

Summary of results

There is a positive result in the first-order setting: $L_1^\star$ is exactly as expressive as one-dimensional first-order temporal logic. See theorem 23.

But in the main, the results are negative. Obviously, $L_1^\star \subseteq L_2^\star \subseteq \cdots$, and $\bigcup_{n \geq 1} L_n^\star = L^\star$. It turns out that $L^\star$ is more expressive than any of the fragments $L_k^\star$ ($k = 1, 2, \ldots$). It follows from this and lemma 22 that even using many-dimensional temporal connectives, we cannot hope for full expressive completeness, even though we have it for one-dimensional connectives (Kamp's theorem 15) in the propositional case. Below we will go into this in more detail.

Expressive completeness for $L_1^\star$

[100, 18] showed that in situations where we have *propositional* expressive completeness, we can derive expressive completeness with respect to $L_1^\star$-formulas. This is one of the few first-order expressive completeness results.

THEOREM 23. *Let $\mathcal{T}$ consist of Until and Since only, and let L be any relational first-order signature.*

1. *Every $\mathcal{T}(L)$-formula is equivalent to some $L_1^\star$-formula over the class of all flows of time.*

2. *For every $L_1^\star$-formula $\varphi(t, x_1, \ldots, x_n)$ with one free $\mathbf{t}$-variable, there is a formula $\delta(x_1, \ldots, x_n)$ of $\mathcal{T}(L)$ that is equivalent to φ over the class of all Dedekind-complete linear flows of time.*

Proof. Part (1) is immediate from lemma 22. The proof of part (2) is by induction on the number of quantifiers $\exists x$ (for any $\mathbf{d}$-variable x) in φ. Let Ψ be the (possibly empty) set of all $L_1^\star$-formulas with at most one free $\mathbf{t}$-variable and with fewer $\mathbf{d}$-quantifiers than φ, and inductively assume the result for all formulas in Ψ. Then φ is built from

1. atomic formulas of the form $u < v$, $u = v$, where u, v have sort $\mathbf{t}$ as usual,

2. atomic $L^\star$-formulas of the form $y = z$ and $P^\star(u, y_1, \ldots, y_k)$,

3. formulas of the form $\exists x\psi(u, x, y_1, \ldots, y_m)$ for $\psi \in \Psi$,

using only the boolean operations and quantifiers over **t**-variables. Each formula in (2) has at most one free **t**-variable, so is of the form $\alpha(u, y_1, \ldots, y_k)$ for some **t**-variable u (where $y_1, \ldots, y_k$ are **d**-variables). Our strategy is to treat the formulas in (2) and (3) as unary relation symbols over the flow of time. Then φ turns into a first-order formula to which we can apply Kamp's theorem. What remains can be dealt with by hand/by the induction hypothesis.

Replace each type (2) subformula $\alpha(u, y_1, \ldots, y_k)$ of φ by $Q_\alpha(u)$, where Q_α is a new unary relation symbol. Similarly, for each ψ as in (3), introduce a unary relation symbol Q_ψ, and replace the subformula $\exists x\psi(u, x, y_1, \ldots, y_m)$ of φ by $Q_\psi(u)$. In this way, we obtain a first-order formula $\varphi'(t)$ involving only time variables, unary relation symbols, equality, and $<$. By Kamp's theorem (theorem 15 above), there is a *propositional US*-formula β, whose atoms are q_α, q_ψ for α, ψ as above, and whose standard translation β^t is equivalent to $\varphi'(t)$ over Dedekind complete time.

By the inductive hypothesis, for each formula $\psi(u, x, y_1, \ldots, y_m) \in \Psi$, there is a $\mathcal{T}(L)$-formula $\gamma_\psi(x, y_1, \ldots, y_m)$ whose standard translation $\gamma_\psi^\star(u, x, y_1, \ldots, y_m)$ is equivalent to ψ over Dedekind complete time. Hence, $(\exists x\gamma_\psi)^\star(u, y_1, \ldots, y_m)$ is equivalent to $(\exists x\psi)(u, y_1, \ldots, y_m)$. Replace the atoms q_ψ in β by $\exists x\gamma_\psi(x, y_1, \ldots, y_m)$, for each such ψ, and similarly replace the atoms q_α in β by $\alpha(y_1, \ldots, y_k)$, for each $\alpha(y_1, \ldots, y_k)$ as in (2). We obtain a $\mathcal{T}(L)$-formula $\delta(x_1, \ldots, x_n)$ which is easily seen to be equivalent to φ over Dedekind complete time. ❑

We proved this result for the connectives Until and Since, but the approach is more general. For example, in part (2) we could capture all linear time by adding the Stavi connectives. We do not know whether there are generalisations to $L_k^\star$ for $k > 1$. The issues of separation and complexity and succinctness of translations in first-order temporal logic remain largely unexplored, though one may expect that the lower bounds for propositional temporal logic will be inherited by the first-order case.

Expressive incompleteness

Without the restriction to $L_1^\star$, the picture is not so rosy. Let us first lay down what we would like.

DEFINITION 24. Let $\mathcal{C}$ be a class of flows of time, L a signature, and $\mathcal{T}$ a set of k-dimensional temporal connectives. We say that $\mathcal{T}(L)$ is *first-order expressively complete over* $\mathcal{C}$ if for every two-sorted first-order $L^\star$-formula $\varphi(t_1, \ldots, t_k, x_1, \ldots, x_n)$, there is a $\mathcal{T}(L)$-formula $\alpha(x_1, \ldots, x_n)$ that is equivalent to φ over $\mathcal{C}$.

It has long been known that connectives that are expressively complete in the propositional case can fail to be so in first-order temporal logic. For example, let L consist of a unary relation symbol Q, and let $\mathcal{T}$ consist of the one-dimensional connectives Until and Since. We know by theorem 15 that $\mathcal{T}$ is propositionally expressively complete over $(\mathbb{N}, <)$ and over the class of finite linear flows. But consider the $L^\star$-sentence

$$\chi \;=\; \exists uv(u \neq v \wedge \forall x(Q^\star(u, x) \leftrightarrow Q^\star(v, x))).$$

This says that the temporal structure contains two identical first-order structures. A somewhat analogous English statement is that all those who get married on some day get divorced on some (other) day too. Now χ is not in $L_1^\star$; and since the standard translation of any $\mathcal{T}(L)$-formula is in $L_1^\star$, we might suspect that χ is not expressible in $\mathcal{T}(L)$. Indeed, it turns out that χ is not equivalent to any $\mathcal{T}(L)$-formula over the flow of time $(\mathbb{Q}, <)$ (proved by Kamp in [113]), nor over $(\mathbb{N}, <)$ or the class of all finite linear flows [4, 3]. So:

THEOREM 25 ([113, 4, 3]). *The connectives Until and Since are not first-order expressively complete over any of* $(\mathbb{Q}, <)$, $(\mathbb{N}, <)$, *or the class of all finite linear flows of time.*

The context of [4, 3] is temporal databases, and other workers in this area have recently been active in temporal expressiveness. We now describe three of their results.

First, [191] showed that there is no finite set of one-dimensional temporal connectives that is expressively complete over temporal structures with dense linear flow of time. The proof was again model-theoretic.

Second, [18] showed that (slightly non-standard versions of) Until and Since cannot express χ (above) in temporal structures with linear flow of time. Whilst this result is implied by that of [113], the proof is novel. It can be summarised as follows. Suppose for contradiction that we had an $L_1^\star$-formula χ_1 equivalent to χ over linear time. Although χ does not involve $<$, χ_1 might: it might use the linear order in some devious way to express χ. But because χ does not involve $<$, the truth value of χ and hence χ_1 in any temporal structure $\mathcal{M}$ with linear flow of time is invariant under replacing the order on the $\mathbf{t}$-sort of $\mathcal{M}$ by any other linear order. So χ_1 may use a linear order, but it doesn't care which order it's given. We can say that χ_1 is 'order-independent over linear time'.

Now it can be checked that $L_1^\star$ has the Craig interpolation property. Using this and order-independence and the fact that being a linear order is first-order definable, we can replace χ_1 by another $L_1^\star$-formula χ_1' in which $<$ does not occur. χ_1' is still equivalent to χ over linear time. But the absence of $<$ makes it relatively easy to exhibit two temporal structures with linear flows of time that differ on χ, but agree on all $L_1^\star$-formulas without $<$ and at most as complex as χ_1'. Hence they agree on χ_1', which is impossible since χ_1' is equivalent to χ over linear time. So over temporal structures with linear flow of time, χ is not expressible in $L_1^\star$, and by theorem 23(1), not expressible by Until and Since.

To use Craig interpolation to remove $<$ from χ_1, it was assumed that χ_1 is equivalent to χ over the class of all linear flows of time. The argument does not appear to work for non-first-order definable classes, such as $\{(\mathbb{N}, <)\}$. It does not exclude that there is some $L_1^\star$-sentence that is not order-independent over a wide range of temporal structures but still is equivalent to χ over $(\mathbb{N}, <)$. ([4, 3] and theorem 23 show that there isn't!)

Third, Toman in [190] again used Craig interpolation to show — roughly — that over linear time, there is no finite expressively complete set of k-dimensional connectives for any finite k. He did this by showing that the $L_k^\star$ $(k = 1, 2, \ldots)$ are not eventually constant in terms of expressive power. Since the standard translations of temporal formulas always lie in some $L_k^\star$, no first-order temporal logic with first-order-definable connectives of any fixed finite dimension can be first-order expressively complete over such a class. As far as we know, it is an open question whether there is a set of temporal connectives of some finite dimension that is first-order expressively complete over non-first-order definable classes such as $(\mathbb{N}, <)$ and the class of all finite linear flows of time.

5 TEMPORAL REASONING

There are several reasoning tasks needed for the applications of temporal logic. The most fundamental of these tasks is deciding which formulas of a logic are valid, or at least enumerating the valid ones, i.e. the formulas which hold at every time point in every structure. Being able to decide validity also allows us (in the obvious way) to decide whether one formula is a logical consequence of any given finite set of formulas. Other important reasoning tasks, which we will not examine in this chapter, include model checking (but see Chapter 17), execution of temporal formulas and synthesis of models from formulas.

Many of the techniques for deciding or enumerating validity in temporal logics are extensions of methods developed for classical logics, or perhaps other modal logics but there are also novel techniques. We shall examine some ten or so different techniques. An algorithm which decides the validity of each formula of a given logic is known as a *decision procedure* for the logic, (an encoded version of) a formula is fed as input to the algorithm and it eventually halts with a (correct) "yes" or "no" answer, indicating whether the formula is valid or not. An algorithm which produces in succession each valid formula of a given logic (as output), and outputs no other formulas, is known as a semi-decision procedure for the logic.

Some techniques are quite general and can be modified to work (as decision or semi-decision procedures) for a wide range of temporal logics but others are quite specific. For a particular logic it is very useful to know of a decision procedure which has the best complexity. But this is not necessarily the end of the story. We might, for example, choose a different algorithm because we want more information such as a counter-model in the case that a formula is not valid. Also, an algorithm with optimal space complexity might not be optimal timewise, and vice versa.

5.1 *Hilbert style axiom systems*

Axiom systems have been presented for logics since the work of Frege [56]. Indeed, they predate semantic formalizations. The Hilbert style approach is the most common sort of axiom system. We assume the reader is familiar with the notions of axioms, rules and proofs in this approach.

In [149], basic axioms and rules are given for the simple propositional temporal logic K_t of F and P over the class of all flows of time. It turns out to be slightly neater to axiomatize the logic with G and H as operators (and F and P as abbreviations), so we will do that. The early work in the area of axiomatizations for the most basic temporal logics was undertaken by Prior, Kripke, Bull, Cochiarella and Burgess and was to a large extent not published. What follows is a distilled mixture that is hard to attribute correctly.

The rules of our system for K_t are modus ponens and temporal generalization,

$$\frac{\alpha, \alpha \to \beta}{\beta} \qquad \frac{\alpha}{G\alpha} \quad \frac{\alpha}{H\alpha}.$$

The axioms (with some redundancy) are all substitution instances of the following:

L0	any propositional tautology		
LF1	$G(\alpha \to \beta) \to (G\alpha \to G\beta)$	LP1	$H(\alpha \to \beta) \to (H\alpha \to H\beta)$
LF2	$G\alpha \to GG\alpha$	LP2	$H\alpha \to HH\alpha$
LF3	$\alpha \to GP\alpha$		
LP3	$\alpha \to HF\alpha$		

A straightforward induction on the lengths of proofs gives us the *soundness* of the axiom system, i.e. that

LEMMA 26. *If there is a proof of α in the above system then α is valid in K_t: i.e. α is true at every time point in every temporal structure.*

The converse is called the *completeness* of the logic.

LEMMA 27. *If α is valid in K_t then there is a proof of α in the above system.*

Proof. By considering negations it is enough to show that if α is consistent (in the system) then it is satisfiable. So suppose that α is consistent. We will build a model of α. There are two well known techniques for doing so: the step by step building of a *perfect chronicle* set out very clearly in [32] and via the *canonical model* as seen in [178] and [27]. We will sketch the latter method here.

By the method of the lemma of Lindenbaum allowing any consistent set of formulas (in such a system) to be extended to a maximal consistent set (MCS), it suffices to show that we can find a model of any MCS Γ.

The model we find is in fact the *canonical model* consisting of exactly all the MCS of our system. That is, let the set of time points be just the set T_0 of all maximal consistent sets.

We put $\Gamma <_0 \Delta$ iff for all $G\alpha \in \Gamma$ we have $\alpha \in \Delta$. It is straightforward to use $LF2$ and $LP2$ to show that this is a transitive relation. Thus $(T_0, <_0)$ is almost a flow of time. Unfortunately, $(T_0, <_0)$ may not be irreflexive. To make an irreflexive frame out of $(T_0, <_0)$ is slightly complicated and the technique of bull-dozing mentioned below is often used. We will skip the details here.

A valuation h_0 can be simply put on this frame via $h_0(p) = \{\Gamma \in T_0 \mid p \in \Gamma\}$.

Now an induction on the construction of formulas gives us a *truth lemma* that for all $\Gamma \in T_0$, for all formulas β, $\beta \in \Gamma$ iff $(T_0, <_0, h_0), \Gamma \models \beta$.

Thus we have our model and we are done. ❑

Note that the proof of the completeness lemma actually establishes that every consistent set of formulas, even an infinite one, is satisfiable. This property is known as the strong completeness of the axiom system for the logic. If we can only show that every consistent formula (or equivalently consistent finite set of formulas) is satisfiable then we only have the weak completeness of a system.

It should also be noted that the axioms for transitivity and the duality of F and P above are in the special form known as Sahlqvist axioms which we discuss shortly below.

1. (a) canonical models, for K_t:

   To axiomatize Prior's propositional language with G and H over some of the other important classes of frames often only involves a sensible choice of an additional few axioms and minor (but sometimes tricky) adjustments to the canonical model method (or equally the chronicle method). We can axiomatize

linearity, density and the lack of end points in this way. See for example, [72] or [64] for details.

For example, to axiomatize the irreflexive, linear frames we can add

L4 $\quad F\alpha \to G(F\alpha \vee \alpha \vee P\alpha)$

or equivalently L4' $\quad (H\alpha \wedge \alpha \wedge G\alpha) \to GH\alpha$

and the past-time mirror image to K_t. The canonical model for this system will not do as it stands for showing satisfiability of consistent formulas as it is not necessarily anti-symmetric or irreflexive. In fact it may well contain *clusters,* which are sets of MCSs such that for every Γ and Δ in the set, $\Gamma <_0 \Delta$ and $\Delta <_0 \Gamma$. In order to make a linear irreflexive structure from the canonical model we can use a technique of [178] called bull-dozing. This means that we replace each cluster by a linear order which is a kind of product of the integers with the cluster.

Note that the case of axiomatizing G and H over rational numbers time can be obtained easily (using some basic model theory and Cantor's characterization of the rationals) with a completeness proof using axioms for linearity, denseness, and lack of end points.

(b) $\mathbb{N}$:

For G and H over the natural numbers $\mathbb{N}$, we can show completeness for the system obtained by adding the following axioms to the system for linear temporal logic [72]:

$D_F \quad F\top$

$Z_F \quad G(G\alpha \to \alpha) \to (FG\alpha \to G\alpha)$

$W_P \quad H(H\alpha) \to H\alpha$

Z_F is a version of Dummett's axiom and allows us to show that between any two time points lie only a finite number of other points.

In attempting to axiomatise this logic one must face its lack of compactness: meaning there is an unsatisfiable set Γ of formulas such that every finite subset of Γ is satisfiable. For example, $\Gamma = \{Fp, FFp, FFFp, \ldots, FG\neg p\}$ is such a set. We are obviously not going to be able to find a point of time in a $\mathbb{N}$-flowed structure at which all the formulas in Γ hold. To show completeness of an axiom system for such a non-compact logic we need slightly different finitary techniques.

See [72] for details of the proof which uses the canonical model and a filtration followed by intricate "surgery". We will meet some of the same ideas and some alternative approaches shortly below when dealing with more expressive logics over natural numbers time.

(c) $\mathbb{R}$:

We will not go into details for the logic with G and H over the reals here, but essentially one uses the system for the rationals plus an axiom for Dedekind completeness

Cont: $\quad HG(H\alpha \to FH\alpha) \to (H\alpha \to G\alpha)$

i.e. that formulas don't highlight gaps in the flow of time. The completeness proof proceeds via the addition of irrational numbers to a rational flowed model with the specification of a maximal consistent set of formulas to hold at each irrational. See [72] for details.

$$
\begin{array}{ll}
\neg Y\alpha \leftrightarrow W\neg\alpha & \neg X\alpha \leftrightarrow N\neg\alpha \\
Y\alpha \rightarrow W\alpha & X\alpha \rightarrow N\alpha \\
\alpha \rightarrow WX\alpha & \alpha \rightarrow NY\alpha \\
W(\alpha \rightarrow \beta) \rightarrow (W\alpha \rightarrow W\beta) & N(\alpha \rightarrow \beta) \rightarrow (N\alpha \rightarrow N\beta) \\
\neg P\alpha \leftrightarrow H\neg\alpha & \neg F\alpha \leftrightarrow G\neg\alpha \\
H(\alpha \rightarrow \beta) \rightarrow (H\alpha \rightarrow H\beta) & G(\alpha \rightarrow \beta) \rightarrow (G\alpha \rightarrow G\beta) \\
H\alpha \rightarrow W\alpha & G\alpha \rightarrow N\alpha \\
H(\alpha \rightarrow W\alpha) \rightarrow (\alpha \rightarrow H\alpha) & G(\alpha \rightarrow N\alpha) \rightarrow (\alpha \rightarrow G\alpha) \\
\alpha S\beta \leftrightarrow \beta \vee (\alpha \wedge Y(\alpha S\beta)) & \alpha U\beta \leftrightarrow \beta \vee (\alpha \wedge X(\alpha U\beta)) \\
P\mathbf{start} & \alpha U\beta \rightarrow F\beta
\end{array}
$$

Figure 3. The axiom schemas for TL

2. U and S:

   If we now consider the more expressive language with U and S then we see two traditions in the work on their axiom systems. For the class of all linear flows we have the Burgess–Xu axioms of [30, 215]. Extra axioms have been added to them to axiomatize U and S over the reals in [157] (interestingly using expressive completeness and techniques with lexicographic sums of orders from [62] and [33]) and the integers and natural numbers in [159] and [199].

   Separately, there has been the development of axiom systems for the discrete (i.e. natural numbers) time logics of PLTL [66] and the past-time version TL [124] which we now describe.

   The rules are modus ponens and temporal generalizations, past and future:

$$
\frac{\alpha, \alpha \rightarrow \beta}{\beta} \qquad \frac{\alpha}{H\alpha} \quad \frac{\alpha}{G\alpha}\ .
$$

   We also allow any substitution instance of any propositional tautology. This can be presented as a rule of inference (called R1 in [124]). Instead we could add all the standard axiom schemas for a complete system for classical propositional logic (see for example, that in [92]) and allow any TL substitution instance of any of them.

   The axioms are all substitution instances of the schemas given in figure 3 and in order to present them at their simplest we have made use of further abbreviations, $W\alpha \equiv \neg Y\neg\alpha$ for "weak yesterday", and $N\alpha \equiv \neg X\neg\alpha$ for "weak tomorrow". So, for example, the first axiom is just $\neg Y\alpha \leftrightarrow \neg Y\neg\neg\alpha$.

   From [124] we have:

   THEOREM 28. $\vdash_{TL}$ *is sound and complete for TL validity.*

   Proof. We give a rough sketch of the completeness proof. This style of proof is appropriate for weak completeness results where we just need to show that a formula which is consistent in the axiom system has a model.

   Suppose we are to find a model of the $\vdash_{TL}$-consistent TL formula ϕ. We will build the model from consistent sets of formulas as in the canonical model construction.

However, the sets will be finite. We specify a particular *closure* set of formulas dependent on ϕ and we make the model from subsets of the closure set. The closure set is usually just the set of subformulas of ϕ and their negations and possibly a few more formulas. In the TL case we also include, for example, $X(\alpha U \beta)$ in the closure set for ϕ iff $\alpha U \beta$ is in the set.

Next we define some of the subsets of the closure set to be *atoms*. The intention is for a set A to be an atom iff there "could" be some model M and some point t in it such that A contains exactly those formulas from the closure set which are true at t in M. The actual definition of an atom is just a set of simple syntactic criteria that rule out sets that could not possibly be satisfied in this sense. For example, we require an atom to include exactly one of α and $\neg\alpha$ for every $\neg\alpha$ in the closure set.

Let W_0 be the set of atoms. To turn W_0 into a graph, we now place a binary relation R_0 on W_0. The intention is for this to represent a successor relation between a time point and the next time. If $(A, B) \in R_0$ then we want it to be possible in some model for a point satisfying A to be followed by a point satisfying B. Again, we actually only use an approximate syntactic test with this aim. For example, if $X\alpha \in A$ then we want $\alpha \in B$.

The construction then proceeds by a series of modifications of (W_0, R_0) to (W_1, R_1), $(W_2, R_2), \ldots$ until we reach a final graph. The modifications are essentially just the removal of parts of W_i to get W_{i+1}. We remove a set of atoms under certain circumstances reflecting the global behaviour of possible models. For example, we look for certain parts of the graph which can not satisfy the eventualities which they contain: for example, an atom contains $\alpha U \beta$ and no R_i-path from here ever reaches an atom containing β.

When this process stabilizes it can be shown that we have ended up with a graph out of which can be read a model for ϕ: we find an infinite path of atoms (some repeated) through the graph. The steps of the process which justify this are themselves justified in terms of the atoms. For the details see [124]. ❑

3. Branching Time

The branching time logic CTL was axiomatized in [42] using a few rather intricate rules to capture inductive reasoning step by step along branches in a tree. The completeness proof is in some ways similar to that for TL above but also follows closely a tableau style decision procedure for CTL.

Axiomatizing CTL* was a much more difficult problem with difficulties raised by its limit closure property whereby any increasing sequence of prefixes of paths is part of one path. There is an axiomatization presented in [163] which uses a special rule motivated by automata-theoretic considerations.

A neater, more traditional axiomatization was shown to be possible in [162, 167] for PCTL*, the expansion of CTL* via past-time temporal operators. Our result is to find a Hilbert system capable of deriving exactly the valid formulas of $\models_P$. We describe that briefly.

There are five rules of inference. These include modus ponens (MP), future temporal generalization (FTG), past temporal generalization (PTG), and branch generalization (BG):

$$MP: \ \frac{\alpha, \alpha \rightarrow \beta}{\beta}\ ; \quad FTG: \ \frac{\alpha}{G\alpha}\ ; \quad PTG: \ \frac{\alpha}{H\alpha}\ ; \quad BG: \ \frac{\alpha}{A\alpha}\ .$$

Finally, there is a special rule, or axiom schema, *atomic non-futurity,* which says that propositional atoms only depend on states and not on fullpaths:
ANF: $p \rightarrow Ap$, for each atomic proposition p.

The axioms include all substitution instances of the following.
TL: the axioms of $\vdash_{TL}$ (including propositional tautologies),
plus a few which say that A acts as an S5 modality:
AK $A(\alpha \rightarrow \beta) \rightarrow (A\alpha \rightarrow A\beta)$
AT $A\alpha \rightarrow \alpha$
AE $E\alpha \rightarrow AE\alpha$
plus two which allow some interaction between modalities:
AX $AX\alpha \rightarrow XA\alpha$
AY[1] $YA\alpha \rightarrow AY\alpha$
and the limit closure axiom from [163]:
LC $AG(E\alpha \rightarrow EX((E\beta)U(E\alpha))) \rightarrow (E\alpha \rightarrow EG((E\beta)U(E\alpha)))$

The completeness proof in [167] is beyond the scope of this chapter but it involves a normal form for TL formulas in which the future time operators are much restricted. That in turn allows us to control the construction of an infinite model by finite sets of formulas from a closure set.

Axiomatic completeness results for many of the variations on logics of historical necessity can be found in [29, 216, 217, 78].

4. Predicate temporal logics are generally so highly undecidable (i.e. not recursively enumerable) that it is impossible to provide for them complete axiom systems [64]. Some rare exceptions include the predicate temporal logic of U and S over linear time [160] and monodic first-order temporal logic [214] which we consider further below.

For some of the temporal logics mentioned above and for many others, there exist alternative axiomatizations using the *Irreflexivity Rule* (IRR) of [60, 29] and variations on this. The IRR allows

$$\frac{q \wedge H(\neg q) \rightarrow \alpha}{\alpha} \text{ provided that the atom } q \text{ does not appear in the formula } \alpha.$$

A short proof shows that this is a valid rule over irreflexive linear frames. It is very useful to have this rule in an axiom system as it allows the construction of something like a canonical model which is irreflexive and in which every point has a unique "name" as the first time at which a particular atom is true. For some of the classes of frames we have seen (including the linear flows) complete axiomatizations can be obtained by using the

[1] A different axiom was mistakenly used in [167].

IRR rule along with the intricate and powerful Sahlqvist persistence theorem [173], [174] which gives us first-order conditions corresponding to validity of temporal formulas in a frame. See [64, chapter 6] for details of this, and of an IRR based axiom system for U and S over the reals.

5.2 Gentzen systems

Hilbert systems are not at all useful for actually carrying out reasoning. The alternative formal reasoning systems proposed by Gentzen in [69] are much better. Gentzen's sequent calculus allows more natural, modular reasoning. A sequent is just a pair consisting of a set of formulas, the premises and a formula which is the conclusion. The calculus allows us to make deductions about the validity of some sequent in terms of the validity of others. The general idea for propositional classical logic is well presented in [92] and [187]. These chapters also describe two popular variants on the sequent calculus, natural deduction and semantic tableaux which we describe below.

Traditional Gentzen style sequent calculi for temporal logics can be found in [145], [183], [114], [203], [108] and [146].

We will not consider sequent calculi in any detail here as the similar tableau systems below are of more interest. Note that for efficient automation, the cut rule is problematic.

5.3 Natural Deduction

Gentzen also introduced the natural deduction calculus for classical logic in [69]. The general idea is that we start at the top with some assumptions, and gradually work down, discharging assumptions as we go until we reach the desired conclusions. The calculus consists of rules of proof and they usually come in pairs governing the introduction or elimination respectively of any given connective. There are close connections to the cut-free sequent calculus and to tableau methods. Proofs can be arranged in a tree shape (Gentzen style) or they can be shaped in a linear way (Jaskowski style) via the use of certain "book-keeping" devices such as boxes and labels. Natural deduction calculi were developed for modal logics by Fitch in [53] with the inclusion of "strict subderivations" representing a jump to another world in a Kripke structure [54].

Indrzejczak [107] contains natural deduction systems for tense logics but a more interesting recent natural deduction calculus for Prior's linear tense logic appears in [109]. This system only uses analytic cut. By having only elimination rules for the temporal operators, by using the structure of labels in the proof to build a model and by generating non-branching proofs, it is able to be used as a decision procedure. There is a natural deduction system for an interval temporal logic in [197].

5.4 Tableaux

Tableaux, which are also closely related to cut-free Gentzen sequent systems [54], are one of the most popular methods for reasoning in modal logics and there has been a substantial amount of work on applying them to temporal logics. They can be presented in an intuitive way, they are often suitable for automated reasoning and it is often not hard to prove complexity results for their use. Tableaux were invented for classical propositional logic in [16] and [89] and they were used for modal logics in [103] and [54].

Standard tableau rules for propositional calculus with connectives just $\neg$ and $\wedge$ are:

$$\frac{\Sigma; \alpha; \neg\alpha}{}, \quad \frac{\Sigma; \neg\neg\alpha}{\Sigma; \alpha}, \quad \frac{\Sigma; \alpha \wedge \beta}{\Sigma; \alpha; \beta}, \quad \frac{\Sigma; \neg(\alpha \wedge \beta)}{\Sigma; \neg\alpha \mid \Sigma; \neg\beta}.$$

The ';' such as in $\Sigma; \alpha \wedge \beta$ represents set union and we omit the braces around singleton sets of formulas. To see whether a formula ϕ is satisfiable, i.e. if $\neg\phi$ is not valid, we use instances of these rules to construct a tree with nodes labelled by sets of formulas (in the language of ϕ). Start with a node labelled by $\{\phi\}$. At any stage there may be a choice of rules to apply. The first rule tells us that we can end a branch and make a leaf node if the label contains a formula and its negation. The second rule says that we can extend a branch by one node if it currently ends with a label $\Sigma \cup \{\neg\neg\alpha\}$ containing a formula $\neg\neg\alpha$ and add a node labelled by $\Sigma \cup \{\alpha\}$. Note that Σ itself may contain $\neg\neg\alpha$ or it may not. The third rule is similar. The fourth rule allows a branch to be formed, a node containing $\neg(\alpha \wedge \beta)$ may have two successor nodes, with the indicated labels.

Certain restrictions on the way we use the rules will guarantee termination of this tableau building process and so we have a decision procedure.

In tableaux for modal logics there are several differences. The labels of nodes can be thought of as representing a set of formulae which we know hold simultaneously in one "possible" world. Some rules can still be used to add propositional consequences and thus tell us more about a particular world. This is called a static rule. However, in the modal setting there are other, transitional rules, which represent movement along the accessibility relation and tell us what is true at a different world. For example, the following rule is appropriate for Prior's F and G over transitive frames:

$$\frac{G\alpha_1, G\alpha_2, \ldots, G\alpha_n, F\beta}{\alpha_1, \alpha_2, \ldots, \alpha_n, G\alpha_1, G\alpha_2, \ldots, G\alpha_n, \beta}$$

Tableau systems are unfortunately not completely straightforward for most temporal logics. The mirror-image past-time modalities and linearity seem to need messy rules. Furthermore, transitivity necessitates rules which can allow indefinite lengths of branches and temporal tableau systems thus often incorporate rules for checking "looping" and specifying that such branches are not closed. There are tableau systems for common propositional temporal logics in [39], [114], [176], [80], [177], [110] as well as the future only case in [79]. The tableau systems for temporal logics in [109] and [129] both have labels on sets of formulas as part of the reasoning process.

An alternative way of managing tableau generation has become popular for the discrete time logics of interest in computing. Instead of arranging the labelled nodes in a tree, a more general graph structure allows repetition of labels to be avoided. This can be more efficient. To use this technique to decide the satisfiability of a formula ϕ say, one must determine from ϕ a finite closure set of formulas of importance to the tableau reasoning. The closure set appropriate for many temporal languages is just the set of subformulas of ϕ and their negations but it may also include a few extra formulas—it depends on the nature of the tableau rules. The tableau initially consists of a graph with one node labelled with each of the maximally propositionally consistent subsets of the closure set. Directed edges (i.e. tableau transitions) are placed between nodes according to syntactic rules which attempt to capture a possible discrete step in time. For example, if $X\alpha$ and $G\beta$ are in the label at one node then a successor node will have to contain both α and β. Depending on exactly the expressiveness of the temporal language, the initial graph

is then subjected to a series of checks to remove unsatisfiable arrangements of nodes. A trivial example is the removal of a node with $X\alpha$ in its label if it has no successor at all. A more complex example is the removal of a node with $F\alpha$ in its label if from it can be found no path of nodes leading to a node labelled with α. For many useful logics, such a process can be defined which eventually terminates with a situation which correctly indicates whether ϕ is satisfiable or not. See [207] and [36] for PLTL, [15] for a fragment of PLTL, [115] for future and past-time operators over natural numbers time, [81] for past-time operators over integers, [42] and [41] for CTL.

For tableaux for first-order temporal logics see [117] and [132]. For intervals see [75]. There are overviews of the use of tableaux for temporal logic in [80] and [40]. See also Chapter 2 in this Handbook.

5.5 Resolution

Resolution for classical logic, predicate and propositional, was first described by Robinson in [170]. The general idea is a complete calculus based on a single rule of inference, the resolution rule, able to be used repeatedly. Formulas need to be in a *clausal form*, which involves rewriting them into conjunctive normal form. We start with a formula, ϕ say, the validity of which is in question, and rewrite $\neg\phi$ as an equivalent formula in the clausal form. Thus $\neg\phi$ is equivalent to the conjunction of a set of clauses, each of which is a disjunction of atomic formulas or their negations. Each application of the resolution rule should add another clause to the set, it being a consequence of two clauses already in the set. If we ever see that the set is inconsistent then we can conclude that $\neg\phi$ is too and so ϕ is valid. The resolution rule allows us to deduce $\alpha \vee \beta$ from $\alpha \vee p$ and $\beta \vee \neg p$ for an atomic formula p.

Resolution for modal logic is discussed in Chapter 4 of this Handbook. Existing methods of temporal resolution are mostly for temporal logics of discrete natural numbers time. The early approach in [34] is for a restricted sublanguage of PLTL without the until operator. There are a whole raft of resolution and transformation rules to be used.

In [1] a complete system for PLTL is presented. It also involves quite a variety of rules, including a "cut" style rule. The system in [202] is a little neater but still involves some initial rewriting followed by a choice of rules and operations to apply.

Perhaps most work has been done on the system first described in [50] and more recently in [52]. The PLTL formula is first translated to an essentially equivalent formula in what is called Separated Normal Form (SNF), which is a clausal form with only a much restricted use of the temporal operators. There are only two resolution rules, the classical one as above and one temporal that at its simplest allows something roughly like $\alpha \rightarrow F\gamma$ to be resolved with $G(\delta \rightarrow \neg\gamma)$ and $G(\delta \rightarrow X\delta)$ to give $\alpha \rightarrow \neg\delta$.

Extensions of these methods to CTL appear in [22], to first order temporal logic in [2] and monodic first-order temporal logic in [37]. Implementations of resolution theorem-provers perform well, even compared to the latest tableau provers [105].

5.6 Automata

Finite state automata are a popular way of carrying out reasoning about temporal logics. We will not give a detailed account here, though, as Chapter 17 in this Handbook is devoted to their use.

Automata have been used to formalize basic notions in computing since the time of Turing but Büchi in [25] was a pioneer of their use with infinite linear structures. Several important temporal reasoning tasks can be translated into tasks requiring manipulation of such automata. For example, a typical test of validity of a temporal formula, may be able to be accomplished by finding an equivalent automaton (to the negation of the formula) and checking to see if that automaton is empty. Here we say that an automaton is equivalent to a formula if the automaton accepts exactly the models of the formula. We say that the automaton is empty if it accepts no structures at all.

The task of finding automata equivalents to temporal formulas (over natural numbers time) is described in [182, 208, 195]. The states of the automaton are basically just nodes of the tableau graph for the formula, its transition relation follows the tableau successor relation and its acceptance criteria capture the need for eventuality formulas (from the closure set) to be fulfilled. Alternative routes for translation proceed via the second order logic $S1S$ (and McNaughton's determinization result [133]) or via the alternating automata of [139] and then on to a Büchi automaton.

The task of determining emptiness of an automaton is described in [45] and [195].

By putting together these results in the right way, a PSPACE decision procedure for PLTL validity can be obtained [195].

Automata are much used to reason about temporal logics on discrete trees—see for example, the Rabin tree automata used in [46, 194, 44] with CTL* but, of course, they are not suited to use with dense time structures unless special discreteness restrictions are imposed on the behaviour of atoms [8], [154].

5.7 Translation into first-order logic

We have seen, in Section 4.1 above, that we can easily translate formulas from many temporal logics into equivalent first-order formulas, via the so-called *standard translation.* If a propositional temporal logic has only connectives with first-order tables then we can do this. Say that temporal formula α using atoms $p_1, \ldots, p_n$ translates to first-order formula $\phi(t)$ with free variables t and predicate symbols $P_1, \ldots, P_n$. The validity of the temporal formula over a class of flows of time is thus equivalent to the validity of $\phi(t)$ over all structures with a frame $(T, <)$ from that class of flows of time and any interpretation for the P_i. That is, we want to assess the validity of $\forall P_1 \ldots \forall P_n \forall x \phi$, a formula from the universal monadic second-order logic over that class of frames.

One straightforward way to decide such a validity question is to use a first-order theorem-prover such as, amongst several other widely-used systems, Otter [111]. Using such a theorem-prover will of course require some syntactic formalization of the class of frames in question and this may or may not be possible. For example, we can axiomatize linearity in order to capture the class of all linear orders. The general validity question here, for first-order formulas, is undecidable but modern theorem-provers are impressively fast in obtaining results for many interesting theorems and non-theorems.

More conclusive decidability results have been obtained for many of the common temporal logics because the universal monadic second-order logic, or even the full monadic second-order logic, over the class of frames in question is itself decidable. There are many such results and we just list a few. Some more details can be found in [58] and [64, chapter 15].

Monadic second-order logic over $(\mathbb{N}, <)$ is known to be decidable [25]: there is an

algorithm to tell whether any given monadic second-order sentence is true on $(\mathbb{N}, <)$ or not. Gurevich [83, 33] showed the same for the universal monadic second-order theory of the class of all linear flows of time and of $(\mathbb{R}, <)$. From these results decidability of any usual temporal logic over those classes follows, where by a usual temporal logic we mean one in which the temporal operators all have first-order tables. The complexity of this decision problem is non-elementary (see, e.g., [153]), but since monadic second-order logic is so expressive, it is useful for proving bare decidability of other logics, by coding them into it.

Over the tree $({}^{<\omega}2, <)$, we sometimes beef up the expressive power by adding two unary function symbols l, r to define the left and right successors of each time point. Regarding ${}^{<\omega}2$ as the set of finite sequences of 0s and 1s, we let $l(s) = s0$ and $r(s) = s1$ for each $s \in {}^{<\omega}2$. The ordering $<$ is definable from l, r alone in monadic second-order logic, since

$$({}^{<\omega}2, <, l, r) \models \forall xy\big(x \leq y \leftrightarrow \forall X[\forall z[X(l(z)) \vee X(r(z)) \to X(z)] \wedge X(y) \to X(x)]\big).$$

So it suffices to work in $({}^{<\omega}2, l, r)$. Monadic second-order logic over this structure is known to be decidable by a theorem of Rabin [152]. It is extremely expressive, and a wide range of other logics can be shown decidable by coding them into it. For example, monadic second-order logic over $(\mathbb{Q}, <)$ can be shown decidable in this way, as can the monadic second-order theory of the class of all countable ω-trees. Decidability of temporal logics over trees was also shown [85] by a translation to a full monadic second-order logic of trees, but one in which the set quantifiers range only over branches of the tree.

Sometimes monadic second-order logic is too strong (for some purposes). For example, it was proved by Shelah in [181] that monadic second-order logic over $(\mathbb{R}, <)$ is undecidable. Shelah's use of the continuum hypothesis in this proof was later eliminated with Gurevich [84]. It follows quite easily that any class of flows of time containing one in which $(\mathbb{R}, <)$ is embeddable has undecidable full monadic second-order theory. So, for example, the monadic second-order theory of the class of all linear flows is undecidable.

5.8 Filtration and the finite model property

Filtration and variants of the finite model property are ideas which can be used together to give theoretical decidability or complexity results but they are not commonly used for practical reasoning.

Filtration is a technique used for, amongst other things, finding a second model of a formula given one model. The second model can often be made to be much smaller and, in particular, of finite size. Filtration is a traditional technique to use with modal logics tracing its origins to the work of Kripke, Lemmon and Segerberg [179]. Its importance for us here is that it can be used to show that if a formula in a temporal language has a model then it has a finite model. If all the formulas of a language have this property then we say that the language has the *finite model property* or *fmp*. As we will see below, the fact of the fmp, or closely related properties, can often be used to give a decision procedure for the logic.

Establishing the fmp is most natural with some non-temporal modal logics. With temporal logics, because of the requirement of irreflexivity, it is often the case that structures have to be infinite in size. Even if some formulas of a particular temporal logic can have finite models, such as with Prior's temporal language over the class of

all linear flows of time, it is often very easy to find a satisfiable formula which only has infinite models, such as say $Fp \wedge G(p \to Fp)$.

Luckily, we can step back from a direct use of the fmp here. Temporal languages can be given alternative semantics on other classes of Kripke structures. For example, Prior's propositional temporal language with F and P over linear flows of time, which we call CL for now, was, in Section 3.4 above, given semantics over any transitive irreflexive structure $(T, <, h)$ and the same clauses even give the formulas semantics on any Kripke structure $(T, <, h)$.

Ono and Nakamura [143] use filtrations and the fmp to assess the complexity of (deciding validity in) CL. Call a Kripke structure $(T, <, h)$ a CL-model of a formula α iff the structure is transitive and totally ordered (i.e. that for every x and y in T, either $x < y$, $y < x$ or $x = y$) and it contains some $t \in T$ such that $(T, <, h), t \models \alpha$. Such a structure generally contains clusters of points, each pair of which are related by $<$, i.e. $(T, <)$ is not generally anti-symmetric and so not a linear order. The main theorem is that a formula α with n temporal operators is satisfiable in an (anti-symmetric) linear model iff it is satisfiable in a finite CL-model containing at most $n + 1$ points.

It had been long established that formulas of Prior's language have linear models iff they have CL-models so the two main new steps in [143] are that if a formula with n operators has a CL-model then it has a finite CL-model and, furthermore, it has a model of size $n + 1$. The first step is via a filtration argument, the second, a clever pruning of a finite model down to a bare minimum of size which still preserves truth of the formula.

As an illustration of filtration let us assume that formula α has a CL-model $(T, <, h)$ and we will sketch how to find a finite model. Suppose $(T, <, h), t_0 \models \alpha$. Now define a binary relation $\sim$ on T by $s \sim t$ iff for all subformulas β of α, we have $(T, <, h), s \models \beta$ iff $(T, <, h), t \models \beta$. This is clearly an equivalence relation so we can define $M = T/\!\sim$ which will be a finite set. Now define an accessibility relation R on M by:

$[s]R[t]$ iff for all subformulas $F\beta$ of α, if $(T, <, h), s \models \neg F\beta$
then $(T, <, h), t \models \neg F\beta$
and $(T, <, h), t \models \neg\beta$; and
for all subformulas $P\beta$ of α, if $(T, <, h), t \models \neg P\beta$
then $(T, <, h), s \models \neg P\beta$
and $(T, <, h), s \models \neg\beta$.

It is not hard to show that R is well-defined, transitive and total. Finally we define a valuation g on (M, R) by $[s] \in g(p)$ iff p is a subformula of α and $s \in h(p)$. This is clearly well-defined. The CL-structure (M, R, g) is called a *filtration* of $(T, <, h)$. Now, a straightforward induction on the construction of β allows us to conclude that $(T, <, h), s \models \beta$ iff $(M, R, g), [s] \models \beta$. Thus we have a finite model of α as required.

We have just established the finite model property for the CL logic. As mentioned, [143] goes further and establish a *bounded model property* by giving an upper bound (in terms of the construction of the formula) on the minimum size of a finite model of the formula.

The finite model property can give us decidability of the logic. If we have an effective proof system, such as a Hilbert-style axiomatization for the logic then there is a way of listing systematically the valid formulas in the logic. If we also have the fmp and an effective way of determining whether finite structures are models of the logic, then we can also systematically consider all the finite structures and eventually list all the satisfiable formulas. Running the two algorithms in parallel thus allows us to eventually

find whether a given formula is valid or whether its negation is satisfiable.

The bounded model property gives a decision procedure itself, with no need of an axiom system, provided there is an effective way of recognizing finite models of the logic. Further, it gives us a complexity measure on a decision procedure. For example, suppose that we want to decide whether α of Prior's language is satisfiable over a linear flow of time. The results of [143] allow us to propose a non-deterministic algorithm in which a CL-model of α is guessed and checked in polynomial time. We can thus conclude that the complexity of satisfiability for CL is NP-complete.

Sistla and Clarke in [184] use another variation of filtrations and finite model properties to prove a PSPACE complexity result for PLTL. The presence of the "Until" operator makes the use of the techniques more intricate but we still identify points in a structure which agree on all the subformulas of the formula to decide. The main result establishes that every satisfiable formula will be satisfied in an ultimately periodic structure with bounds on its period and on the length of the non-periodic prefix. See Chapter 17 for details.

5.9 Other modal methods

We have only a little space to mention a powerful series of results by Wolter [209, 212, 211, 210, 213], extending to temporal logic deep work of Chagrov, Fine, Zakharyaschev, and others about modal logics above K4. The results offer the potential to extend modal reasoning (and other) methods to temporal logics with the Priorean F and P in a uniform and systematic way.

In [213], Wolter remarks that unlike in modal logic, the focus of work in temporal logic has been on specific systems such as the theory of $(\mathbb{N}, <)$, $(\mathbb{Z}, <)$, or $(\mathbb{R}, <)$. This is inappropriate for building a mathematical theory of the entirety of temporal logics, where "we are usually not interested in the properties of a specific system but in results which show why a specific system has a property by extracting concepts which allow to map out the boundaries of that property." Wolter investigates the preservation of properties (such as Kripke completeness, finite model property, decidability, finite axiomatizability, and canonicity) when we 'temporalize' a modal logic by adding a converse diamond (in the same sense that P is the converse of F). The frequent failure of the finite model property in temporal logics is the chief obstacle to such transfers, even when the corresponding modal logic does have the fmp. In fact, [209] proves that the temporalization of a modal logic $\Lambda \supseteq$ K4 whose frames are closed under taking cofinal subframes has the fmp precisely when the class of frames for Λ is elementary. [213] shows that the temporalization is decidable if Λ is finitely axiomatizable, and gives further results on transfer of fmp, Kripke completeness, and decidability, though a full characterization of when decidability transfers is left open. One of the main methods used is a replacement for the fmp. It involves replacing points of a finite frame by 'blocks', which in simple cases can be irreflexive points, copies of $(\mathbb{N}, >)$, and so on, but can also be general frames. Related lattice and algebraic results (for example, on canonicity) are presented in [212, 211].

5.10 Mosaics

Mosaics were used to prove decidability of certain theories of cylindric relativized set algebras in [140, 141] and have been used since quite generally in algebraic logic and

modal logic. Mosaics are small potential pieces of a model: in the temporal logic case, a small piece of a temporal structure. The main idea is to show that the existence of a model of a given formula is equivalent to the existence of a finite set of these small pieces, satisfying certain coherency conditions. The set of mosaics should also satisfy certain saturation conditions. This gives us a decision procedure, and, intuitively, a systematic procedure to check the theoremhood of a certain formula. It is possible to view some older approaches such as in [66] and [30] as specialisations of mosaics. However, recently the mosaic method has been applied to prove new decidability, Hilbert-style axiomatizability and complexity results for various modal and temporal logics: see, e.g., [90], [135], [161], [201], [129] and [166].

In [129], temporal mosaics are described which are appropriate for reasoning with Prior's logic with F and P over linear flows of time. To give a flavour of the mosaic method we present some of the details here.

Mosaics, for a particular purpose, are defined with respect to a fixed closure set of formulas, say X, which is closed under taking subformulas and such that if $\alpha \in X$ then either so is $\neg\alpha$ or $\alpha = \neg\beta$ and $\beta \in X$. Let X be such a closure set in the propositional temporal language with F and P (and $\neg$ and $\wedge$) and some countable set of atoms.

First the definition of a mosaic, via the coherency conditions:

DEFINITION 29. A mosaic is a pair (A, B) of subsets of X such that:

1 A and B are maximally propositionally consistent;

2 $G\alpha \in A$ implies α and $G\alpha$ in B;

3 similarly for $H\alpha \in B$.

Next, the so called saturation conditions on the whole set of mosaics:

DEFINITION 30. A set M of mosaics is a saturated set of mosaics (SSM) iff:

1 if $(A, B) \in M$ and $F\alpha \in B$ then there is a mosaic $(A', B') \in M$ such that $B = A'$ and $\alpha \in B'$;

2 if $(A, B) \in M$ and $F\alpha \in A$ then either $\alpha \in B$ or $F\alpha \in B$ or there are mosaics (A', B') and (A'', B'') in M such that $A = A'$, $\alpha \in B' = A''$ and $B'' = B$;

3 and 4 similarly for $P\alpha$.

The main lemma shows that satisfiability of a set Γ of formulas is equivalent to (either there being a one point model of Γ or) there being a saturated set of mosaics M containing one mosaic (A, B) with $\Gamma \subseteq A$ or $\Gamma \subseteq B$. The mosaics from an SSM can be re-used and glued together in a very intuitive manner to build a model.

From this main lemma follows a completeness proof for a Hilbert-style axiomatization (using the set of all formulas as X), a completeness result for a tableau system, and (using a finite set X) decidability and complexity results. The decision procedure is simply a systematic cropping of the set of all mosaics (for that X) down to a saturated set.

In [166], which contains the first detailed suggestions on how mosaics might be used for reasoning with dense time temporal logics, this general approach is used to determine the complexity of deciding validity of formulas of the temporal logic with "Until" (but not "Since") over the class of all linear flows of time. The mosaics and their coherency and saturation conditions are a little different but they still correspond to pieces of a model defined by two points in time. Here, finding an SSM is shown to be equivalent to the existence of a winning strategy for one player in a two-player game played with mosaics. The search for a winning strategy can be arranged into a search through a tree

of mosaics which we can proceed through in a depth-first manner. By establishing limits on the depth of the tree (a polynomial in terms of the length of the formula) and on the branching factor (exponential) we can ensure that we have a PSPACE algorithm as we only need to remember a small fixed amount of information about all the previous siblings of a given node.

The idea is again used in [168] to give a PSPACE complexity result for deciding valid formulas in the propositional temporal language with strict U and S over real-numbers flows.

5.11 Monodic fragments of first-order temporal logic

To end our survey, we look again very briefly at *first-order temporal logic.* The development of this subject has been slower than for propositional temporal logic, possibly in part because of the poor computational behaviour of first-order temporal logic. Unpublished results of Lindström and Scott in the 1960s showed that even very weak fragments of it can be (sometimes highly) undecidable. More results like this have been proved over the years. In [100, theorems 2, 73], for example, the two-variable monadic fragment and the two-variable guarded fragment of first-order temporal logic were shown to be non-recursively enumerable over $(\mathbb{N}, <)$ and $(\mathbb{Z}, <)$, by encoding a recurrent tiling problem. The first-order temporal logic with U and S can be axiomatised over linear time [160], but this is one of few positive results. The 'reason' for the high complexity is the product-like structure of models of first-order temporal logic. They combine a flow of time with a first-order structure, and the interaction between the two leads to very high expressive power.

Recently, however, some interesting *decidable fragments* of first-order temporal logic have come to light. One way to obtain such fragments is by limiting the interaction in the logic between time and the first-order structure. Lemma 22 gives us a clue how to do it. There, we saw that the standard translations of (one-dimensional) temporal formulas satisfied the restriction that any subformula $\exists x\varphi$, for a domain variable x, had at most one free time variable. This reflected the fact that temporal formulas are evaluated at a single time point. If we aim for a similar restriction on quantified temporal variables as well, we are on the way to obtaining decidable fragments — the so-called *monodic fragments* of [100].

DEFINITION 31. A formula α of first-order temporal logic is said to be *monodic* if every subformula of α beginning with a temporal connective has at most one free variable.

The Barcan formula, $\forall x G\varphi(x) \leftrightarrow G\forall x\varphi(x)$, gives an example of a monodic formula. Here, G applies to $\varphi(x)$ (one free variable) and to $\forall x\varphi(x)$ (no free variables). If φ is monodic then the above Barcan formula will be too. Another example, of Chomicki and Toman from databases, is to 'list all persons who have been unemployed between jobs':

$$P\exists y\ \mathit{WorksFor}(x, y) \wedge \neg\exists y\ \mathit{WorksFor}(x, y) \wedge F\exists y\ \mathit{WorksFor}(x, y).$$

A non-example is the sentence $\forall xy((R(x, y) \rightarrow GR(x, y)) \wedge (\neg R(x, y) \rightarrow G\neg R(x, y)))$ expressing rigidity of a binary relation. Here, the G applies to formulas with two free variables. So the sentence is not monodic.

The full monodic fragment is in general *undecidable,* since it contains first-order logic. So we need to make further restrictions to obtain decidable fragments. Some restrictions that have been found to work for many linear flows of time are:

1. The signature must have at most only relation symbols and constants (no function symbols).

2. The 'first-order part' of the formulas must come from a decidable fragment of first-order logic. Examples include the monadic fragment (i.e., the signature can have only unary relation symbols), the two-variable fragment, and the various guarded fragments.

3. Equality (=) must not occur [38], although this restriction is not necessary if we choose guarded fragments in (2) [96].

We can now obtain several decidable fragments of first-order temporal logic, depending on what fragment of first-order logic we pick in (2). For example, we could select the monodic monadic fragment, the monodic guarded fragment, etc. Which we choose may depend on what application we have in mind for our logic. A formula of one of these fragments can describe the evolution over time of only one domain object. But at any one time, the full power of the chosen fragment of first-order logic is available.

Under the above restrictions, it was shown in [100] that satisfiability of monodic formulas may be determined by combining an algorithm to decide the fragment chosen in (2) with a second algorithm to decide *monadic second-order logic* over the ambient flow of time. This procedure can be made to work whenever the flow of time is linear and its monadic second-order theory is decidable. This is true for many common flows of time, such as $(\mathbb{N}, <)$, $(\mathbb{Z}, <)$, and $(\mathbb{Q}, <)$: see Section 3.2.

The key idea in the proofs is the *quasimodel.* A quasimodel is essentially the result of filtrating a temporal structure over its first-order domain; the flow of time is left untouched. (See Section 5.8 and Chapter 1 for filtration.) Roughly, a quasimodel for a given monodic sentence φ written with the connectives U and S, say, contains the following ingredients:

1. A *type* is a description of a single domain point, in terms of which subformulas ψ of φ with one free variable are true at it. So a type is a subset of the set of subformulas of φ with one free variable together with their negations.

   The quasimodel contains, for each time t, a set Σ_t of types. It is required that Σ_t be the set of types of the elements of some first-order structure. For this purpose, maximal subformulas of ψ of the form $U(\alpha, \beta)$ and $S(\alpha, \beta)$ are regarded as atomic. In this way, only the *first-order part* of the description is kept.

2. The types in Σ_t are there to describe the domain elements at time t. To recover some information about the types of a single domain element over time, the quasimodel also contains a set of what are called *runs.* A run ρ is simply a choice of type $\rho(t) \in \Sigma_t$ at each time t, subject to the coherence conditions

   - for any subformula of φ of the form $U(\alpha, \beta)$, *which by monodicity has at most one free variable,* and for any time t, we have $U(\alpha, \beta) \in \rho(t)$ iff there is a time $u > t$ with $\alpha \in \rho(u)$ and $\beta \in \rho(v)$ for all times v strictly between t and u,
   - a mirror-image condition for S.

   The quasimodel must have *enough runs.* Formally for each t, each type in Σ_t must be the value at t of some run in the quasimodel.

It turns out that a monodic sentence has a genuine model iff it has a quasimodel, and that the existence of a quasimodel can be expressed in monadic second-order logic and hence is frequently decidable.

THEOREM 32 ([100, 96]). *Let $(T, <)$ be a flow of time with decidable monadic second-order theory (examples include $(\mathbb{N}, <)$, $(\mathbb{Z}, <)$, and $(\mathbb{Q}, <)$). Then for any of the following sets of sentences, the problem of whether a sentence from the set has a model with flow of time $(T, <)$ is decidable:*

1. *the monodic sentences with only unary relation symbols and no equality,*
2. *the monodic sentences involving at most two variables and no equality,*
3. *the monodic sentences with 'first-order part' in the guarded, loosely guarded, packed, or clique-guarded fragment of first-order logic (see Chapter 5 and [100] for precise definitions).*

We can easily generalise this to any first-order-definable class $\mathcal{C}$ of linear flows of time. This covers the class of all linear flows, the class of discrete linear flows, the class of dense linear flows, etc.

The case of $(\mathbb{R}, <)$ is special and interesting. The monadic second-order theory of $(\mathbb{R}, <)$ is undecidable, so the method described above does not work directly. As yet, it is not known whether monodic fragments are decidable over this flow. However, monodic fragments can be shown decidable over $(\mathbb{R}, <)$ if we restrict to *temporal structures with finite first-order domains* [100]. This is an interesting case in its own right (in databases, for example), and it yields, by reduction, decidability for finite domains over $(\mathbb{N}, <)$, $(\mathbb{Z}, <)$ $(\mathbb{Q}, <)$, linear flows, finite linear flows, and so on.

The complexity of these decision procedures is non-elementary, but in fact the satisfiability problem for monodic fragments can be shown to be typically the maximum of EXPSPACE and the complexity of the underlying first-order fragment — see [98] and [97], the second of which extends the mosaic methods in [168] to first-order temporal logic. Over $(\mathbb{N}, <)$, the full unrestricted monodic fragment has been axiomatised [214], and tableau and resolution decision procedures have been developed [117, 37, 116].

For *branching time,* we may want to use the logic CTL*, with temporal connectives Until and the path quantifier A (see Section 3.7). However, there are serious problems with this. Over ω-trees (trees with all branches isomorphic to $(\mathbb{N}, <)$: see Section 2.1), and even replacing Until by the weaker $F_{\leq}$ of Section 3.4, the one-variable fragment of CTL* is undecidable [102]; this is certainly monodic and its first-order part is decidable. Decidable fragments of CTL* can be found by requiring monodicity and that the first-order part of sentences comes from a decidable fragment of first-order logic, as above, but additionally, either that any subformula beginning with a temporal connective other than 'tomorrow' must be a sentence [102], or alternatively that first-order quantification is only applied to state formulas [14]. Some applications of monodic fragments are discussed in [101].

6 CONCLUSION

We have tasted a wide range of topics from half a century of development of temporal logic. The reader will now hopefully appreciate that temporal logics come in many forms

and that motivations from computing or linguistic applications and philosophical, theoretical or mathematical interests have driven temporal logic research in many disparate directions.

We must admit that owing to space limitations, we have not done justice to some topics (consider interval based temporal logics for example), and other topics have not even been mentioned — the reader may like to follow up on temporal bisimulations in [118] for example.

Other general references and surveys of the whole field, some of historical interest, include [21, 39, 51, 64, 67, 72, 126, 127, 142, 156, 185, 193, 192, 200].

The reader may also begin to appreciate that this is an active area of research and that there is still much to do. Particular directions of current interest include combinations of temporal and other modal logics, and the search for practical logics (or fragments of logics) which are amenable to affordable and usable reasoning techniques.

ACKNOWLEDGEMENTS

Much of the second author's work on this chapter was undertaken while he was employed by Murdoch University, Western Australia. Both the authors wish to thank V. Goranko and J. van Benthem for careful reading of the chapter and numerous insightful suggestions for improvement.

BIBLIOGRAPHY

[1] M. Abadi and Z. Manna. Nonclausal Temporal Deduction. *Lecture Notes in Computer Science*, 193:1–15, 1985.

[2] M. Abadi and Z. Manna. Non-clausal deduction in first-order temporal logic. *J. ACM*, 37:279–317, 1990.

[3] S. Abiteboul, L. Herr, and J. van den Bussche. Temporal connectives versus explicit timestamps in temporal query languages. In J. Clifford and A. Tuzhilin, editors, *Recent Advances in Temporal Databases*, pages 43–57. Springer, 1995.

[4] S. Abiteboul, L. Herr, and J. van den Bussche. Temporal versus first-order logic in query temporal databases. In *ACM Symposium on Principles of Database Systems*, pages 49–57, Montreal, Canada, 1996.

[5] M. Adler and N. Immerman. An $n!$ lower bound on formula size. *ACM Transactions on Computational Logic*, 4:296–314, 2003.

[6] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, November 1983.

[7] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.

[8] R. Alur and T. A. Henzinger. Real-time logics: Complexity and expressiveness. *Information and Control*, 104:35–77, 1993.

[9] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49:672–713, 2002.

[10] A. Amir. Expressive completeness failure in branching time structures. *Journal of Computer and System Sciences*, 34(1):27–42, 1982.

[11] A. Amir. *Functional Completeness in Tense Logic*. PhD thesis, Bar-Ilan University, Ramat-Gan, Israel, 1982.

[12] A. Amir. Separation in nonlinear time models. *Information and Control*, 66:177–203, 1985.

[13] A. Amir and D. M. Gabbay. Preservation of expressive completeness in temporal models. *Information and Computation*, 72:66–83, 1987.

[14] S. Bauer, I. Hodkinson, F. Wolter, and M. Zakharyaschev. On non-local propositional and weak monodic quantified CTL*. *Journal of Logic and Computation*, 14:3–22, 2004.

[15] M. Ben-Ari. *Mathematical Logic for Computer Science*. Prentice Hall, 1993.

[16] E. Beth. Semantic entailment and formal derivability. *Mededelingen der Koninklijke Nederlandse Akad. van Wetensch*, 18:309–342, 1955.

[17] C. Bettini, S. Jajodia, and X. S. Wang. *Time Granularities in Databases, Data Mining, and Temporal Reasoning.* Springer-Verlag, 2000.

[18] N. Bidoit, S. de Amo, and L. Segoufin. Order independent temporal properties. *J. Logic Computat.*, 14:277–298, 2004.

[19] P. Blackburn and M. Tzakova. Hybrid languages and temporal logic. *Logic Journal of the IGPL*, 7(1):27–54, 1999.

[20] A. Bochman. Concerted instant-interval temporal semantics I: Temporal ontologies. *Notre Dame J. of Formal Logic*, 31(3):403–414, 1990.

[21] L. Bolc and A. Szalas, editors. *Time and Logic: A Computational Approach.* University College London Press, 1995.

[22] A. Bolotov and M. Fisher. A Resolution Method for CTL Branching-Time Temporal Logic. In *Proceedings of TIME-97 the Fourth International Workshop on Temporal Representation and Reasoning*, pages 20–27, Daytona Beach, Florida, May 1997. IEEE Computer Society Press.

[23] P. Bouyer, F. Chevalier, and N. Markey. On the expressiveness of TPTL and MTL. In R. Ramanujam and S. Sen, editors, *Proc. FSTTCS 2005*, LNCS vol 3821, pp. 432–443, Springer, 2005.

[24] H. Bowman, H. Cameron, and S. Thompson. Specification and prototyping of structured multimedia documents using interval temporal logic. In H. Barringer, M. Fisher, D. M. Gabbay, and G. Gough, editors, *Advances in Temporal Logic*, pages 435–453. Kluwer, 2000.

[25] J. R. Büchi. On a decision method in restricted second order arithmetic. In *Logic, Methodology, and Philosophy of Science: Proc. 1960 Intern. Congress*, pages 1–11. Stanford University Press, 1962.

[26] R. A. Bull. An approach to tense logic. *Theoria*, 36:282–300, 1970.

[27] R. A. Bull and K. Segerberg. Basic modal logic. In D. M. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic II*, pages 1–88. Reidel, 1984.

[28] J. P. Burgess. Logic and time. *J. Symbolic Logic*, 44:566–582, 1979.

[29] J. P. Burgess. Decidability for branching time. *Studia Logica*, 39:203–218, 1980.

[30] J. P. Burgess. Axioms for tense logic I: 'Since' and 'Until'. *Notre Dame J. Formal Logic*, 23(2):367–374, 1982.

[31] J. P. Burgess. Axioms for tense logic II. Time periods. *Notre Dame Journal of Formal Logic*, 23(4):375–383, 1982.

[32] J. P. Burgess. Basic tense logic. In D. M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, vol II: Extensions of Classical Logic*, pages 89–113. Reidel, Dordrecht, 1984.

[33] J. P. Burgess and Y. Gurevich. The decision problem for linear temporal logic. *Notre Dame J. Formal Logic*, 26(2):115–128, 1985.

[34] A. Cavalli and L. Fariñas del Cerro. A Decision Method for Linear Temporal Logic. In R. E. Shostak, editor, *Proceedings of the 7th International Conference on Automated Deduction*, volume 170 of *Lecture Notes in Computer Science*, pages 113–127. Springer Verlag, 1984.

[35] E. M. Clarke and E. Emerson. Synthesis of synchronization skeletons for branching time temporal logic. In *Proc. IBM Workshop on Logic of Programs, Yorktown Heights, NY*, pages 52–71. Springer, Berlin, 1981.

[36] E. M. Clarke and H. Schlingloff. Model checking. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning, volume II*, chapter 24, pages 1635–1790. Elsevier Science, 2001.

[37] A. Degtyarev, M. Fisher, and B. Konev. Monodic Temporal Resolution. In *Proceedings of Conference on Automated Deduction (CADE)*, pages 397–411. Springer, July 2003.

[38] A. Degtyarev, M. Fisher, and A. Lisitsa. Equality and monodic first-order temporal logic. *Studia Logica*, 72:147–156, 2002.

[39] E. Emerson. Temporal and modal logics. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 995–1072. Elsevier, 1990.

[40] E. Emerson. Automated temporal reasoning for reactive systems. In F. Moller and G. Birtwistle, editors, *Logics for Concurrency*, pages 41–101. Springer Verlag, 1996.

[41] E. Emerson and E. M. Clarke. Using branching time temporal logic to synthesise synchronisation skeletons. *Sci. of Computer Programming*, 2:241–266, 1982.

[42] E. Emerson and J. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *J. Comp and Sys. Sci*, 30(1):1–24, 1985.

[43] E. Emerson and J. Halpern. 'Sometimes' and 'not never' revisited: on branching versus linear time. *J. ACM*, 33:151–178, 1986.

[44] E. Emerson and C. Jutla. Complexity of tree automata and modal logics of programs. In *29th IEEE Foundations of Computer Science, Proceedings*, pages 328–337. IEEE, 1988.

[45] E. Emerson and C. Lei. Modalities for model checking: branching time strikes back. In *Proc. 12th ACM Symp. Princ. Prog. Lang.*, pages 84–96, 1985.

[46] E. Emerson and A. Sistla. Deciding full branching time logic. *Information and Control*, 61:175 – 201, 1984.

[47] K. Etessami and T. Wilke. An until hierarchy and other applications of an Ehrenfeucht–Fraïssé game for temporal logic. *Information and Computation*, 160:88–108, 2000.

[48] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge.* The MIT Press, 1995.

[49] M. Finger and M. Reynolds. Imperative history: two-dimensional executable temporal logic. In U. Reyle and H.-J. Ohlbach, editors, *Logic, Language and Reasoning: Essays in Honour of Dov Gabbay*, pages 81–107. Kluwer, 1999.

[50] M. Fisher. A resolution method for temporal logic. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 99–104, August 1991.

[51] M. Fisher. Temporal Representation and Reasoning. In F. van Harmelen, B. Porter, and V. Lifschitz, editors, *Handbook of Knowledge Representation*, volume 2 of *Foundations of Artificial Intelligence.* Elsevier, 2006.

[52] M. Fisher, C. Dixon, and M. Peim. Clausal Temporal Resolution. *ACM Transactions on Computational Logic*, 2(1):12–56, January 2001.

[53] Fredric Brenton Fitch. *Symbolic Logic.* Ronald Press Company, 1952.

[54] M. Fitting. *Proof methods for modal and intuitionistic logics.* Reidel, 1983.

[55] M. Franceschet and A. Montanari. Branching within time: an expressively complete and elementarily decidable temporal logic for time granularity. *Research on Language and Computation*, 1:229–263, 2003.

[56] G. Frege. *Conceptual Notation (Begriffschrift), and related articles.* Oxford : Clarendon Press, 1972. English translation, T. Bynum, ed.

[57] T. French and M. Reynolds. A sound and complete proof system for QPTL. In N-Y. Suzuki and F. Wolter, editors, *Advances in Modal Logic, volume 4*, pages 229–240. Kings College Publications, 2003.

[58] D. M. Gabbay. *Investigations in Modal and Tense Logics with Applications to Problems in Philosophy and Linguistics*, volume 92 of *Synthese.* D. Reidel, Dordrecht, 1976.

[59] D. M. Gabbay. Expressive functional completeness in tense logic (preliminary report). In U. Monnich, editor, *Aspects of Philosophical Logic*, pages 91–117. Reidel, Dordrecht, 1981.

[60] D. M. Gabbay. An irreflexivity lemma with applications to axiomatizations of conditions on tense frames. In U. Monnich, editor, *Aspects of Philosophical Logic*, pages 67–89. Reidel, Dordrecht, 1981.

[61] D. M. Gabbay. Declarative past and imperative future: Executable temporal logic for interactive systems. In B. Banieqbal, H. Barringer, and A. Pnueli, editors, *Proceedings of Colloquium on Temporal Logic in Specification, Altrincham, 1987*, volume 398 of *Lecture Notes in Computer Science*, pages 67–89. Springer-Verlag, 1989.

[62] D. M. Gabbay and I. Hodkinson. An axiomatisation of the temporal logic with until and since over the real numbers. *Journal of Logic and Computation*, 1(2):229–260, 1990.

[63] D. M. Gabbay, I. Hodkinson, and M. Reynolds. Temporal expressive completeness in the presence of gaps. In J. Oikkonen and J. Väänänen, editors, *Logic Colloquium '90*, volume 2 of *Lecture Notes in Logic*, pages 89–121. Springer-Verlag, 1993.

[64] D. M. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal logic: mathematical foundations and computational aspects, Vol. 1.* Clarendon Press, Oxford, 1994. Also see volume 2, 2000.

[65] D. M. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications.* Volume 148 of Studies in Logic and the Foundations of Mathematics, North-Holland, 2003.

[66] D. M. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the Temporal Analysis of Fairness. In *Proceedings of the Seventh ACM Symposium on the Principles of Programming Languages*, pages 163–173, Las Vegas, Nevada, January 1980. ACM Press, NY.

[67] A. Galton. *Temporal logics and their applications.* Academic Press, 1987.

[68] J. W. Garson. Quantification in modal logic. In D. M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, pages 249–307. D. Reidel, 1984.

[69] G. Gentzen. Untersuchungen über das logische Schliessen. *Math. Zeitschrift*, 39:176–210, 405–431, 1934.

[70] K. Gödel. A remark about the relationship between relativity theory and idealistic philosophy. In P. A. Schlipp, editor, *Albert Einstein: Philosopher-Scientist*, pages 555–561. Cambridge University Press, 1970.

[71] R. Goldblatt. Diodorean modality in Minkowski space-time. *Studia Logica*, 39:219–236, 1980.

[72] R. Goldblatt. *Logics of Time and Computation*, volume 7 of *CSLI Lecture Notes.* Center for the Study of Language and Information, Stanford University, 2nd edition, 1992.

[73] R. Goldblatt. *Mathematics of Modality*, volume 43 of *CSLI Lecture Notes.* Center for the Study of Language and Information, Stanford University, 1993.

[74] V. Goranko. Computation tree logics and temporal logics with reference pointers. *Journal of Applied Non-classical Logics*, 10(3):221–242, 2000.

[75] V. Goranko, A. Montanari, and G. Sciavicco. A general tableau method for propositional interval temporal logics. In *Proc. Internat. Conference on Automated Reasoning with Analytic Tableaux and Related Methods (Tableaux 2003)*, pages 102–116. Springer-Verlag, 2003.

[76] V. Goranko, A. Montanari, and G. Sciavicco. Propositional interval neighborhood logics. *Journal of Universal Computer Science*, 9(9):1137–1167, 2003.

[77] V. Goranko, A. Montanari, and G. Sciavicco. A road map of propositional interval temporal logics and duration calculi. *Journal of Applied Non-classical Logics*, 14(1):11–56, 2004.

[78] V. Goranko and G. van Drimmelen. Complete axiomatization and decidability of alternating-time temporal logic. *Theoretical Computer Science*, 353:93–117, 2006.

[79] R. Goré. Cut-free sequent and tableau systems for propositional diodorean modal logics. *Studia Logica*, 53(3):433–458, 1994.

[80] R. Goré. Tableau methods for modal and temporal logics. In M. D'Agostino, D. M. Gabbay, R. Hähnle, and J. Posegga, editors, *Handbook of Tableau Methods*, pages 297–396. Kluwer Academic Publishers, 1999.

[81] G. D. Gough. Decision Procedures for Temporal Logic. Master's thesis, Department of Computer Science, University of Manchester, October 1984. Also University of Manchester, Department of Computer Science, Technical Report UMCS-89-10-1.

[82] M. Grohe and N. Schweikardt. The succinctness of first-order logic on linear orders. *Logical Methods in Computer Science*, 1(1:6):1–25, 2005. `www.lmcs-online.org`.

[83] Y. Gurevich. Elementary properties of ordered abelian groups. *Algebra and Logic*, 3:5–39, 1964. (Russian; an English version is in Trans. Amer. Math. Soc. 46 (1965), 165–192).

[84] Y. Gurevich and S. Shelah. Monadic theory of order and topology in ZFC. *Annals of Math. Logic*, 23:179–198, 1982.

[85] Y. Gurevich and S. Shelah. The decision problem for branching time logic. *J. of Symbolic Logic*, 50:668–681, 1985.

[86] T. Hafer and W. Thomas. Computation tree logic CTL^* and path quantifiers in the monadic theory of the binary tree. In *Proceedings of International Colloquium on Automata, Languages and Programming*, volume 267 of *Lecture Notes in Computer Science*, pages 269–279. Springer-Verlag, 1987.

[87] J. Halpern and Y. Shoham. A propositional modal logic of time intervals. In *Proceedings, Symposium on Logic in Computer Science*, pages 279–292. IEEE, Boston, 1986.

[88] J. Halpern and M. Vardi. The complexity of reasoning about knowledge and time. *JCSS*, 38:195–237, 1989.

[89] J. Hintikka. Form and content in quantification theory. *Acta Philosophica Fennica*, 8:7–55, 1955.

[90] R. Hirsch, I. Hodkinson, M. Marx, S. Mikulás, and M. Reynolds. Mosaics and step-by-step (Remarks on 'A modal logic of relations'). In E. Orłowska, editor, *Logic at Work*, volume 24 of *Studies in Fuzziness and Soft Computing*, pages 158–167. Physica-Verlag, Heidelberg/New York, 1998.

[91] Y. Hirschfield and A. Rabinovich. Logics for real time: Decidability and complexity. *Fundamenta Informaticae*, 62:1–28, 2004.

[92] W. Hodges. Elementary predicate logic. In D. M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, pages 1–132. D. Reidel, Dordrecht, 1983. Also 2nd edition Kluwer 2001.

[93] I. Hodkinson. Finite H-dimension does not imply expressive completeness. *J. Philosophical Logic*, 23:535–573, 1994.

[94] I. Hodkinson. Expressive completeness of Until and Since over Dedekind complete linear time. In A. Ponse, M. de Rijke, and Y. Venema, editors, *Modal logic and process algebra*, volume 53 of *Lecture notes*, pages 171–185. CSLI, 1995.

[95] I. Hodkinson. On Gabbay's temporal fixed point operator. *Theoretical Computer Science*, 139:1–25, 1995.

[96] I. Hodkinson. Monodic packed fragment with equality is decidable. *Studia Logica*, 72:185–197, 2002.

[97] I. Hodkinson. Complexity of monodic packed fragment over linear and real time. *Annals of Pure and Applied Logic*, 138:94–125, 2006.

[98] I. Hodkinson, R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyaschev. On the computational complexity of decidable fragments of first-order linear temporal logics. In Reynolds and Sattar [169], pages 91–98.

[99] I. Hodkinson and A. Simon. The k-variable property is stronger than H-dimension k. *J. Philosophical Logic*, 26(1):81–101, 1997.

[100] I. Hodkinson, F. Wolter, and M. Zakharyaschev. Decidable fragments of first-order temporal logics. *Ann. Pure Appl. Logic*, 106:85–134, 2000.

[101] I. Hodkinson, F. Wolter, and M. Zakharyaschev. Monodic fragments of first-order temporal logics: 2000–2001 A.D. In R. Nieuwenhuis and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence and Reasoning*, volume 2250 of *LNAI*, pages 1–23. Springer-Verlag, 2001.

[102] I. Hodkinson, F. Wolter, and M. Zakharyaschev. Decidable and undecidable fragments of first-order branching temporal logics. In *Proc. 17th IEEE Symposium on Logic in Computer Science (LICS)*, pages 393–402. IEEE Inc., 2002.

[103] G. Hughes and M. Cresswell. *An Introduction to Modal Logic*. Methuen, London, 1968.

[104] E. V. Huntington. A new set of postulates for betweenness, with proof of complete independence. *Transactions of the American Mathematical Society*, 26:257–282, 1924.

[105] U. Hustadt and B. Konev. TRP++2.0: A temporal resolution prover. In *Proceedings of 19th International Conference on Automated Deduction (CADE)*, LNCS, pages 274–278. Springer, July/August 2003.

[106] N. Immerman and D. Kozen. Definability with bounded number of bound variables. In *LICS87, Proceedings of the Symposium on Logic in Computer Science, Ithaca, New York*, pages 236–244, Washington, 1987. Computer Society Press.

[107] A. Indrzejczak. Natural deduction system for tense logics. *Bulletin of the Section of Logic*, 23(4):173–179, 1994.

[108] A. Indrzejczak. Multiple sequent calculus for tense logics. In H. Wansing and F. Wolter, editors, *ICTL 2000, Proceedings*, pages 93–104. University of Leipzig, 2000.

[109] A. Indrzejczak. A labelled natural deduction system for linear temporal logic. *Studia Logica*, 75:345–376, 2003.

[110] G. Jaeger, A. Heuerding, S. Schwendimann, F. Achermann, P. Balsiger, P. Brambilla, H. Zimmermann, M. Bianchi, K. Guggisberg, and W. Heinle. LWB–The Logics Workbench 1.0. `www.lwb.unibe.ch`, University of Berne, Switzerland.

[111] J. A. Kalman. *Automated Reasoning with Otter*. Rinton Press, 2001.

[112] H. Kamp. *Tense logic and the theory of linear order*. PhD thesis, University of California, Los Angeles, 1968.

[113] H. Kamp. Formal properties of "now". *Theoria*, 37:237–273, 1971.

[114] R. Kashima. Cut-free sequent calculi for some tense logics. *Studia Logica*, 53:119–135, 1994.

[115] Y. Kesten, Z. Manna, H. McGuire, and A. Pnueli. A decision algorithm for full propositional temporal logic. In *The Proceedings of Computer Aided Verification (CAV)*, volume 697 of *Lecture Notes in Computer Science*, pages 97–109. Springer, 1997.

[116] B. Konev, A. Degtyarev, and M. Fisher. Handling equality in monodic temporal resolution. In M. Vardi and A. Voronkov, editors, *Logic for Programming and Automated Reasoning (LPAR 2003)*, volume 2850 of *Lecture Notes in Artificial Intelligence*, pages 214–228, Berlin, 2003. Springer-Verlag.

[117] R. Kontchakov, C. Lutz, F. Wolter, and M. Zakharyaschev. Temporalizing tableaux. *Studia Logica*, 76:91–134, 2004.

[118] N. Kurtonina and M. de Rijke. Bisimulations for temporal logic. *Journal of Logic, Language and Information*, 6:403–425, 1997.

[119] P. Ladkin. *The Logic of Time Representation*. PhD thesis, University of California, Berkeley, 1987.

[120] R. Ladner and J. Reif. The logic of distributed protocols. In J. Halpern, editor, *Theoretical Aspects of Reasoning about Knowledge (Proc. 1986 Conference)*, pages 207–222. Kaufman, 1986.

[121] L. Lamport. The temporal logic of actions. Technical Report 79, DEC Systems Research Center, Palo Alto, Calif., USA, 1991.

[122] F. Laroussinie, N. Markey, and Ph. Schnoebelen. Temporal logic with forgettable past. In *Proc. 17th Annual IEEE Symposium on Logic in Computer Science (LICS'02)*, pages 383–392. IEEE Computer Society Press, 2002. Available at `www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/LMS-lics2002.pdf`.

[123] F. Laroussinie and Ph. Schnoebelen. Specification in CTL+past for verification in CTL. *Information and Computation*, 156:236–263, 2000.

[124] O. Lichtenstein, A. Pnueli, and L. Zuck. The glory of the past. In R. Parikh, editor, *Logics of Programs* (*Proc. Conf. Brooklyn USA 1985*), volume 193 of *Lecture Notes in Computer Science*, pages 196–218. Springer-Verlag, Berlin, 1985.

[125] C. Lutz, D. Walther, and F. Wolter. Quantitative temporal logics: PSPACE and below. In *12th Internat. Symp. on Temporal Representation and Reasoning (TIME 2005)*, pages 138–146. IEEE Computer Society, 2005.

[126] Zohar Manna and Amir Pnueli. *The Temporal Logic of Reactive and Concurrent Systems, Vol. 1: Specifications*. Springer, New York, 1992.

[127] Zohar Manna and Amir Pnueli. *The Temporal Logic of Reactive and Concurrent Systems, Vol. 2: Safety*. Springer, New York, 1995.

[128] M. Marx. Conditional XPath, the first order complete XPath dialect. In Alin Deutsch, editor, *Proc. 23rd ACM Symposium on Principles of Database Systems*, pages 13–22. ACM, 2004.

[129] M. Marx, S. Mikulás, and M. Reynolds. The mosaic method for temporal logics. In R. Dyckhoff, editor, *Automated Reasoning with Analytic Tableaux and Related Methods, Proceedings of International Conference, TABLEAUX 2000, St. Andrews, Scotland, July 2000*, LNAI 1847, pages 324–340. Springer, 2000.

[130] M. Marx and M. Reynolds. Undecidability of compass logic. *J. Logic and Computation*, 9(6):897–914, 1999.

[131] M. Marx and Y. Venema. *Multi-Dimensional Modal Logic*. Applied Logic Series No.4 Kluwer Academic Publishers, 1997.

[132] H. W. McGuire. *Two Methods for Checking Formulas of Temporal Logic.* PhD thesis, Department of Computer Science, Stanford University, June 1995. Stanford Computer Science Technical Reports CS-TR-95-1551.

[133] R. McNaughton. Testing and generating infinite sequences by finite automata. *Information and Control*, 9:521–530, 1966.

[134] A. R. Meyer. The inherent computational complexity of theories of ordered sets. In *Proc. International Cong. of Mathematicians (vol. 2)*, pages 477–482, August 1974.

[135] S. Mikulás. Taming first-order logic. *Journal of the IGPL*, 6(2):305–316, 1998.

[136] F. Moller and A. Rabinovich. On the expressive power of CTL^*. In *Proc. Logic in Computer Science*, pages 360–369. IEEE Press, 1999.

[137] A. Montanari, A. Peron, and A. Policriti. Extending Kamp's theorem to model time granularity. *Journal of Logic and Computation*, 12(4):641–678, 2002.

[138] B. Moszkowski. *Executing Temporal Logic Programs.* Cambridge University Press, 1986.

[139] D. Muller, A. Saoudi, and P. Schupp. Weak alternating automata give a simple explanation of why most temporal and dynamic logics are decidable in exponential time. In *3rd IEEE Symposium on Logic in Computer Science, Proceedings*, pages 422–427. IEEE, 1988.

[140] I. Németi. *Free Algebras and Decidability in Algebraic Logic.* PhD thesis, Hungarian Academy of Sciences, Budapest, 1986. In Hungarian.

[141] I. Németi. Decidable versions of first order logic and cylindric-relativized set algebras. In L. Csirmaz, D. M. Gabbay, and M. de Rijke, editors, *Logic Colloquium '92*, pages 171–241. CSLI Publications, 1995.

[142] P. Øhstrom and P. F. V. Hasle. *Temporal Logic: From Ancient Ideas to Artificial Intelligence*, volume 57 of *Studies in Linguistic and Philosophy.* Dordrecht: Kluwer Academic Publishers, 1996.

[143] H. Ono and A. Nakamura. On the size of refutation Kripke models for some linear modal and tense logics. *Studia Logica*, 39:325–333, 1980.

[144] J. Ostroff. Composition and refinement of discrete real-time systems. Technical report, DCS, York University, Ontario, Canada, 1998. CS-1998-10.

[145] B. Paech. Gentzen-systems for propositional temporal logics. In E. Börger, H. Kleine Büning, and M. Richter, editors, *Computer Science Logic '88, Proceedings*, volume 385 of *LNCS*, pages 240–253, 1988.

[146] R. Pliuškevičius. Deduction-Based Decision Procedure for a Clausal Miniscoped Fragment of FTL. In *Proceedings of the First International Joint Conference on Automated Reasoning (IJCAR)*, volume 2083 of *Lecture Notes in Artificial Intelligence*, pages 107–120. Springer, 2001.

[147] A. Pnueli. The temporal logic of programs. In *Proceedings of the Eighteenth Symposium on Foundations of Computer Science*, pages 46–57, 1977. Providence, RI.

[148] B. Poizat. Deux ou trois choses que je sais de L_n. *Journal of Symbolic Logic*, 47:641–658, 1982.

[149] A. Prior. *Time and Modality.* Oxford University Press, 1957.

[150] A. Prior. *Past, Present and Future.* Oxford University Press, 1967.

[151] A. Prior. *Papers on Time and Tense.* Oxford University Press, 1968. New edition 2003.

[152] M. O. Rabin. Decidability of second order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, 141:1–35, 1969.

[153] M. O. Rabin. Decidable theories. In J. Barwise, editor, *Handbook of Mathematical Logic*, pages 595–629. North-Holland, 1977.

[154] A. Rabinovich. On the decidability of continuous time specification formalisms. *Journal of Logic and Computation*, 8:669–678, 1998.

[155] H. Reichenbach. *Elements of Symbolic logic.* Macmillan, New York, 1947.

[156] N. Rescher and A. Urquhart. *Temporal Logic.* Springer-Verlag, 1971.

[157] M. Reynolds. An axiomatization for Until and Since over the reals without the IRR rule. *Studia Logica*, 51:165–193, May 1992.

[158] M. Reynolds. Axiomatisation and decidability of F and P in cyclical time. *Journal of Philosophical Logic*, 23:197–224, 1994.

[159] M. Reynolds. Axiomatizing U and S over integer time. In D. M. Gabbay and H.-J. Ohlbach, editors, *Temporal Logic*, volume 827 of *Lecture Notes in Artificial Intelligence*, pages 117–132. Springer-Verlag, 1994.

[160] M. Reynolds. Axiomatising first-order temporal logic: Until and since over linear time. *Studia Logica*, 57:279–302, 1996.

[161] M. Reynolds. A decidable logic of parallelism. *Notre Dame Journal of Formal Logic*, 38:419–436, 1997.

[162] M. Reynolds. More past glories. In *Fifteenth Annual IEEE Symposium on Logic in Computer Science (LICS'2000), Santa Barbara, California, USA, June 26-28, 2000*, pages 229–240. IEEE, 2000.

[163] M. Reynolds. An axiomatization of full computation tree logic. *J. Symbolic Logic*, 66(3):1011–1057, 2001.

[164] M. Reynolds. Continuous temporal models. In M. Stumptner, D. Corbett, and M. Brooks, editors, *Australian Artificial Intelligence Conference, AI'01, Adelaide, Australia, December 11-14, 2001, Proceedings*, volume 2256 of *Lecture Notes in A.I.*, pages 414–425. Springer-Verlag, 2001.

[165] M. Reynolds. Axioms for branching time. *Journal of Logic and Computation*, 12:679–697, 2002.

[166] M. Reynolds. The complexity of the temporal logic with until over general linear time. *Journal of Computer and System Sciences*, 66:393–426, 2003.

[167] M. Reynolds. An axiomatization of PCTL*. *Information and Computation*, 201: 72–119, 2005.

[168] M. Reynolds. The complexity of the temporal logic over the reals, submitted. Version available at http://www.csse.uwa.edu.au/∼mark/research/Online/CORT.htm.

[169] M. Reynolds and A. Sattar, editors. *Proc. TIME–ICTL*. IEEE Press, 2003.

[170] J. A. Robinson. A Machine–Oriented Logic Based on the Resolution Principle. *Journal of the ACM*, 12(1):23–41, January 1965.

[171] J. G. Rosenstein. *Linear orderings*. Academic Press, New York, 1982.

[172] R. Rosner and A. Pnueli. A choppy logic. In *First Annual IEEE Symposium on Logic in Computer Science LICS'86*, pages 306–313. IEEE Computer Society Press, 1986.

[173] H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logic. In S. Kanger, editor, *Proceedings of the Third Scandinavian Logic Symposium, Uppsala, 1973*, pages 110–143, Amsterdam, 1975. North Holland.

[174] G. Sambin and V. Vaccaro. A new proof of Sahlqvist's theorem on modal definability and completeness. *J. Symbolic Logic*, 54:992–999, 1989.

[175] B-H. Schlingloff. Expressive completeness of temporal logic over trees. *Journal of Applied Non-classical Logics*, 2:157–180, 1992.

[176] P. Schmitt and J. Goubault-Larrecq. A tableau system for linear-time temporal logic. In *TACAS 1997*, pages 130–144, 1997.

[177] S. Schwendimann. A New One-Pass Tableau Calculus for PLTL. In Harrie de Swart, editor, *Proceedings of Tableaux 98*, volume 1397 of *Lecture Notes in Artificial Intelligence*, pages 277–291. Springer-Verlag, 1998.

[178] K. Segerberg. Modal logics with linear alternative relations. *Theoria*, 36:301–322, 1970.

[179] K. Segerberg. An essay in classical modal logic. *Filosofiska Studier*, 13, 1971. Uppsala University.

[180] V. B. Shehtman. Modal logics of domains on the real plane. *Studia Logica*, 42:63–80, 1983.

[181] S. Shelah. The monadic theory of order. *Annals of Mathematics*, 102:379–419, 1975.

[182] R. Sherman, A. Pnueli, and D. Harel. Is the interesting part of process logic uninteresting: a translation from PL to PDL. *SIAM J. on Computing*, 13:825–839, 1984.

[183] T. Shimura. Cut-free systems for the modal logic S4.3 and S4.3GRZ. *Reports on Mathematical Logic*, 25:57–73, 1991.

[184] A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *J. ACM*, 32:733–749, 1985.

[185] C. Stirling. Modal and temporal logics. In *Handbook of Logic in Computer Science, volume 2 (Background: Computational Structures)*, pages 477–563. Clarendon Press, Oxford, 1992.

[186] L. J. Stockmeyer. *The complexity of decision problems in automata theory and logic*. PhD thesis, Dept. of Electrical Engineering, M.I.T., 1974. Project MAC Tech. Report 133.

[187] G. Sundholm. Systems of deduction. In D. M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, volume 1*, pages 133–188. D. Reidel, Dordrecht, 1983.

[188] A. U. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass, editors. *Temporal Databases: Theory, Design and Implementation*. Benjamin/Cummings, 1993.

[189] R. Thomason. Combinations of tense and modality. In D. M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, Vol II: Extensions of Classical Logic*, pages 135–165. Reidel, Dordrecht, 1984.

[190] D. Toman. On incompleteness of multi-dimensional first-order temporal logics. In Reynolds and Sattar [169], pages 99–106.

[191] D. Toman and D. Niwinski. First-order queries over temporal databases inexpressible in first-order logic. In P. Apers, G. Gardarin, and M. Bouzeghoub, editors, *Advances in Database Technology EDBT'96*, volume 1057 of *Lecture Notes in Computer Science*, pages 307–324. Springer-Verlag, 1996.

[192] J. van Benthem. *The Logic of Time*. Reidel, Dordrecht, 1983.

[193] J. van Benthem. Temporal logic. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming: Epistemic and Temporal Reasoning, volume 4*, pages 241–351. Oxford University Press, Oxford, 1995.

[194] M. Vardi and P. Wolper. Yet another process logic. In *Logics of Programs*, volume 164 of *LNCS*, pages 501–512. Springer-Verlag, 1984.

[195] M. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115:1–37, 1994.

[196] Y. Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame J. Formal Logic*, 31:529–547, 1990.

[197] Y. Venema. A modal logic for chopping intervals. *Journal of Logic and Computation*, 1(4):453–476, 1991.

[198] Y. Venema. Relational games. In H. Andréka, J. D. Monk, and I. Németi, editors, *Algebraic logic*, volume 54 of *Colloq. Math. Soc. J. Bolyai*, pages 695–718. North-Holland, Amsterdam, 1991.

[199] Y. Venema. Completeness by completeness: Since and Until. In M. de Rijke, editor, *Diamonds and Defaults*, pages 349–358. Kluwer Academic Publishers, Dordrecht, 1993.

[200] Y. Venema. Temporal logic. In L. Goble, editor, *The Blackwell Guide to Philosophical Logic*, pages 203–223. Blackwell Publishers, Malden, USA, 2001.

[201] Y. Venema and M. Marx. A modal logic of relations. In E. Orłowska, editor, *Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa*, pages 124–167. Physica-Verlag, Heidelberg/New York, 1999.

[202] G. Venkatesh. A Decision Method for Temporal Logic based on Resolution. In *Proceedings 5th Foundations of Software Technologies and Theoretical Computer Science Conference 1985*, volume 206 of *Lecture Notes in Computer Science*, pages 272–289, 1986.

[203] H. Wansing. *Displaying Modal Logics*. Kluwer, Dordrecht, 1999.

[204] P. Wolper. Specification and synthesis of communicating processes using an extended temporal logic. In *Ninth Symp on Principles of Programming Languages*, pages 20–33, Albuquerque, New Mexico, 1982.

[205] P. Wolper. *Synthesis of communicating processes from temporal logic specifications*. PhD thesis, Stanford University, 1982.

[206] P. Wolper. Temporal logic can be more expressive. *Information and Computation*, 56(1-2):72–99, 1983.

[207] P. Wolper. The tableau method for temporal logic: an overview. *Logique et Analyse*, 28:110–111, June–Sept 1985.

[208] P. Wolper, M. Vardi, and A. Sistla. Reasoning about infinite computation paths. In *Proc. 24th ACM Symp. on Foundations of Comp. Sci.*, pages 185–194, 1983.

[209] F. Wolter. The finite model property in tense logic. *Journal of Symbolic Logic*, 60:757–774, 1995.

[210] F. Wolter. A counterexample in tense logic. *Notre Dame Journal of Formal Logic*, 37:167–173, 1996.

[211] F. Wolter. Properties of tense logics. *Mathematical Logic Quarterly*, 42:481–500, 1996.

[212] F. Wolter. Tense logic without tense operators. *Mathematical Logic Quarterly*, 42:145–171, 1996.

[213] F. Wolter. Completeness and decidability of tense logics closely related to logics above $K4$. *Journal of Symbolic Logic*, 62:131–158, 1997.

[214] F. Wolter and M. Zakharyaschev. Axiomatizing the monodic fragment of first-order temporal logic. *Ann. Pure. Appl. Logic*, 118:133–145, 2002.

[215] Ming Xu. On some U, S-tense logics. *J. of Philosophical Logic*, 17:181–202, 1988.

[216] A. Zanardo. Axiomatization of 'Peircean' branching-time logic. *Studia Logica*, 49:183–195, 1990.

[217] A. Zanardo. A complete deductive system for Since-Until branching time logic. *J. Philosophical Logic*, 1991.

[218] A. Zanardo. Branching-time logic with quantification over branches: the point of view of modal logic. *J. Symbolic Logic*, 61:1–39, 1996.

[219] A. Zanardo. First-order and second-order aspects of branching-time semantics. Technical Report N.03, 17 febbraio 2003, Preprint Dipartimento di Matematica Universita di Padova, 2003. Presented at HPLMC (2nd Internat. Workshop on the History and Philosophy of Logic, Mathematics and Computation, Donostia - San Sebastian (Spain), 7-9 November 2002). Avail. from `www.math.unipd.it/~azanardo/Sanseb.pdf`.

[220] C. Zhou and M. R. Hansen. *A Formal Approach to Real-Time Systems*. EATCS Series Monographs in Theoretical Computer Science. Springer, 2004.

[221] C. Zhou, C. Hoare, and A. Ravn. A calculus of durations. *Information Processing Letters*, 40(5):269–276, 1991.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

12

MODAL MU-CALCULI

Julian Bradfield and Colin Stirling

1 Introduction . . . 722
2 Contextual background . . . 722
2.1 Modal logics in program verification . . . 723
2.2 Precursors to modal mu-calculus . . . 724
2.3 The small model property . . . 725
3 Syntax and semantics of modal mu-calculus . . . 726
3.1 Fixpoints as recursion . . . 727
3.2 Approximating fixpoints and μ as 'finitely' . . . 728
3.3 Syntax of $L\mu$ . . . 729
3.4 Semantics of $L\mu$ . . . 730
3.5 Examples . . . 731
3.6 Fixpoint regeneration and the 'fundamental semantic theorem' . . . 732
3.7 Modal equation systems . . . 734
4 Expressive power . . . 735
4.1 CTL and friends as fragments of $L\mu$ . . . 735
4.2 Bisimulation and tree model property . . . 736
4.3 $L\mu$ and automata . . . 737
4.4 $L\mu$ and games . . . 739
5 Decidability of satisfiability . . . 740
5.1 The aconjunctive fragment . . . 740
5.2 Towards automata . . . 741
5.3 Alternating parity automata . . . 742
5.4 Automaton normal form . . . 744
6 Complete axiomatization . . . 745
7 Alternation . . . 746
8 Bisimulation invariance . . . 748
8.1 $L\mu$ and MSOL . . . 748
8.2 Multi-dimensional $L\mu$ and Ptime . . . 749
8.3 Bisimulation quantifiers and interpolation . . . 750
9 Generalized mu-calculi . . . 751
9.1 $L\mu$ with past . . . 751
9.2 Least fixpoint logic . . . 751
9.3 Finite variable fixpoint logics . . . 752
9.4 Guarded fragments . . . 753
9.5 Inflationary mu-calculus . . . 753

1 INTRODUCTION

Modal mu-calculus is a logic used extensively in certain areas of computer science, but also of considerable intrinsic mathematical and logical interest. Its defining feature is the addition of inductive definitions to modal logic; thereby it achieves a great increase in expressive power, and an equally great increase in difficulty of understanding. It includes many of the logics used in systems verification, and is quite straightforward to evaluate. It also provides one of the strongest examples of the connections between modal and temporal logics, automata theory and the theory of games.

In this chapter, we survey a range of the questions and results about the modal mu-calculus and related logics. For the most part, we remain at survey level, giving only outlines of proofs; but in places, determined partly by our own interests and partly by our sense of which problems have been — or had been — the longest-standing thorns in the side of the mu-calculus community, we go into more detail.

We start with an account of the historical context leading to the introduction of the modal mu-calculus. Then we define the logic formally, describe some approaches to gaining an intuitive understanding of formulae, and establish the main theorem about the semantics. Following that, we discuss how the modal mu-calculus has the tree model property and relates to some other temporal logics, to automata and to games. Next, an account of decidability is given — this is one of the thorns, at least for those who find automata prickly. We then consider briefly completeness, bisimulation invariance and then the concept of fixpoint alternation, which plays a part in several interesting questions about the logic. Finally, we look at some generalizations of the logic.

Before proceeding to the content of the chapter, we take this opportunity to thank Yde Venema and Johan van Benthem for extensive and helpful comments on drafts of this chapter.

Notation: $L\mu$ means the modal mu-calculus, considered as a logical language (not as a theory). In general, the notation follows as much as possible the standards for this book, but because $L\mu$ is mostly studied in a setting with rather different traditions, and because we also need to notate several other concepts, we have made some compromises. Few of these should cause any difficulty, but let us note the following. Since $\rightarrow$ is often used to represent the transition relation in models (alias the accessibility relation from modal logic), we use $\Rightarrow$ rather than $\rightarrow$ for boolean implication. Structures, frames and models for $L\mu$ are usually viewed as transition systems, and so are usually called $\mathfrak{T}$ with state space $\mathfrak{S}$. States within systems (i.e. worlds in the language of modal logic) are typically s, t, whereas p, q, r are states in an automaton. Hence we write atomic propositions with capital $P, Q, \ldots$ rather than $p, q, \ldots$, and similarly variables ranging over sets of states are written X, Y.

2 CONTEXTUAL BACKGROUND

The modal mu-calculus comes not from the philosophical tradition of modal logic, but from the application of modal and temporal logics to program verification. In this section, we outline the historical context for $L\mu$.

2.1 Modal logics in program verification

The application of modal and temporal logics to programs is part of a line of program verification going back to the 1960s and program schemes and Floyd–Hoare logic. Originally the emphasis was on *proof*: Floyd–Hoare logic allows one to make assertions about programs, and there is a proof system to verify these assertions. This line of work has, of course, continued and flourished, and today there are highly sophisticated theories for proving properties of programs, with equally sophisticated machine support for these theories. However, the use of proof systems has some disadvantages, and one hankers after a more purely algorithmic approach to simple problems. One technique was pioneered by Manna and Pnueli [48], who turned program properties into questions of satisfiability or validity in first order logic, which can then be attacked by means that are not just proof-theoretic; this idea was later applied by them to linear temporal logics.

During the 1970s, the theory of program correctness was extended by investigating more powerful logics, and studying them in a manner more similar to the traditions of mathematical logic. A family of logics which received much attention was that of dynamic logics, which can be seen as extending the ideas of Hoare logic [57]. Dynamic logics are modal logics, where the different modalities correspond to the execution of different programs — the formula $\langle\alpha\rangle\phi$ is read as 'it is possible for α to execute and result in a state satisfying ϕ'. The programs may be of any type of interest; the variety of dynamic logic most often referred to is a propositional language in which the programs are built from atomic programs by regular expression constructors; henceforth, Propositional Dynamic Logic, PDL, refers to this logic. PDL is interpreted with respect to a model on a Kripke structure, formalizing the notion of the global state in which programs execute and which they change — each point in the structure corresponds to a possible state, and programs determine a relation between states giving the changes effected by the programs.

Once one has the idea of a modal logic defined on a Kripke structure, it becomes quite natural to think of the finite case and write programs which just check whether a formula is satisfied. This idea was developed in the early 80s by Clarke, Emerson, Sistla and others. They worked with a logic that has much simpler modalities than PDL — in fact, it has just a single 'next state' modality — but which has built-in temporal connectives such as 'until'. This logic is CTL, and it and its extensions remain some of the most popular logics for expressing properties of systems.

Meanwhile, the theory of process calculi was being developed in the late 70s, most notably by Milner [50]. An essential component was the use of labelled Kripke structures ('labelled transition systems') as a raw model of concurrent behaviour. An important difference between the use of Kripke structures here and their use in program correctness was that the states are the behaviour expressions themselves, which model concurrent systems, and the labels on the accessibility relation (the transitions) are simple actions (and not programs). The criterion for behavioural equivalence of process expressions was defined in terms of observational equivalence (and later in terms of bisimulation relations). Hennessy and Milner introduced a primitive modal logic in which the modalities refer to actions: $\langle a\rangle\phi$ 'it is possible to do an a action and then have ϕ be true', and its dual $[a]\phi$ 'ϕ holds after every a action'. Together with the usual boolean connectives, this gives Hennessy–Milner logic [31], HML, which was introduced as an alternative exposition of observational equivalence. However, as a logic HML is obviously inadequate to express many properties, as it has no means of saying 'always in the future' or other temporal connectives — except by allowing infinitary conjunction. Using an infinitary

logic is undesirable both for the obvious reason that infinite formulae are not amenable to automatic processing, and because infinitary logic gives much more expressive power than is needed to express temporal properties.

In 1983, Dexter Kozen published a study of a logic that combined simple modalities, as in HML, with fixpoint operators to provide a form of recursion. This logic, the modal mu-calculus, has become probably the most studied of all temporal logics of programs. It has a simple syntax, an easily given semantics, and yet the fixpoint operators provide immense power. Most other temporal logics, such as the CTL family, can be seen as fragments of $L\mu$. Moreover, this logic lends itself to transparent model-checking algorithms.

Another 'root' to understanding modal logics is the work in the 60s on automata over infinite words and trees by Büchi [13] and Rabin [60]. The motivation was decision questions of monadic second-order logics. Büchi introduced automata as a normal form for such formulae. This work founded new connections to logic and automata theory. Later it was realised that modal logics are merely sublogics of appropriate monadic second-order logic, and that the automata normal forms provide a very powerful framework within which to study properties of modal logic. Moreover, automata theoretic algorithms often provide very efficient ways to solve problems (such as model-checking) in modal logic – see Chapter 17 of this Handbook. A further development was the use of games by Gurevich and Harrington [30] as an alternative to automata.

There is also an older game-theoretic tradition due to Ehrenfeucht and Fraïssé, for understanding the expressive power of logics. These techniques are also applicable within process calculi. For instance bisimulation equivalence can be naturally rendered as such a game, and expressivity of modal logics can be understood using game-theoretic techniques.

2.2 Precursors to modal mu-calculus

HML, Hennessy–Milner Logic [31], is a primitive modal logic of action. The syntax of HML has, in addition to the boolean operators, a modality $\langle a \rangle \phi$, where a is a process *action*. A structure for the logic is just a labelled transition system. Atomic formulas of the logic are the constants $\top$ and $\bot$. The meaning of $\langle a \rangle \phi$ is 'it is possible to do an a-action to a state where ϕ holds'. The formal semantics is given in the obvious way by inductively defining when a state of a transition system, or a state of a process, has a property; for example, $s \models \langle a \rangle \phi$ iff $\exists t. s \xrightarrow{a} t \wedge t \models \phi$. We may also add some notion of variable or atomic proposition to the logic, in which case we provide a valuation which maps a variable to the set of states at which it holds. The expressive power of HML in this form is quite weak: obviously a given HML formula can only make statements about a given finite number of steps into the future. HML was introduced not so much as a language to express properties, but rather as an aid to understanding process equivalence: two (image-finite) processes are equivalent iff they satisfy exactly the same HML formulae. To obtain the expressivity desired in practice, we need stronger logics.

The logic PDL, Propositional Dynamic Logic [57, 25], as mentioned above, is both a development of Floyd–Hoare style logics, and a development of modal logics. Recently, it has been used as a basis for description logics and logics of information. PDL is an extension of HML in the circumstance that the action set has some structure. There is room for variation in the meaning of action, but in the standard logic, a program is considered to have a number of *atomic actions*, which in process algebraic terms are just

process actions, and α is allowed to be a regular expression over the atomic actions: a, $\alpha;\beta$, $\alpha\cup\beta$, or α^*. We may consider atomic actions to be uninterpreted atoms; but in the development from Floyd–Hoare logics, one would see the atomic actions as, for example, assignment statements in a **while** program.

PDL enriches the labels in the modalities of HML. An alternative extension of HML is to include further modalities. The branching time logic CTL, Computation Tree Logic [14], can be described in this way as an extension of HML, with some extra 'temporal' operators which permit expression of liveness and safety properties. For the semantics we need to consider 'runs' of a system. A run from an initial state or process s_0 is a sequence $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots$ which may have finite or infinite length; if it has finite length then its final process is a 'sink' process which has no transitions. A run $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots$ has the property $\phi\,\mathbf{U}\,\psi$, 'ϕ until ψ', if there is an $i \geq 0$ such that $s_i \models \psi$ and for all $j : 0 \leq j < i$, $s_j \models \phi$.

$$\begin{array}{ccccccccc} s_0 & \xrightarrow{a_1} & s_1 & \xrightarrow{a_2} & \ldots & s_i & \xrightarrow{a_{i+1}} & \ldots \\ \models & & \models & & \ldots & \models & & \\ \phi & & \phi & & & \psi & & \end{array}$$

The formula $\mathbf{F}\phi = (\top\,\mathbf{U}\,\phi)$ means 'ϕ eventually holds' and $\mathbf{G}\phi = \neg(\top\,\mathbf{U}\,\neg\phi)$: '$\phi$ always holds'. For each 'temporal' operator such as $\mathbf{U}$ there are two modal variants, a strong variant ranging over all runs of a process and a weak variant ranging over some run of a process. We preface a strong version with $\forall$ and a weak version with $\exists$. If HML is extended with the two kinds of until operator the resulting logic is a slight but inessential variant of CTL (CTL does not in its standard form mention action labels in modalities). The formal semantics is given by inductively defining when a state (process) has a property. For instance $s \models \forall[\phi\,\mathbf{U}\,\psi]$ iff every run of s has the property $\phi\,\mathbf{U}\,\psi$.

CTL has variants and enrichments such as CTL* [24] and ECTL* [70]. These allow free mixing of path operators and quantifiers: for example, the CTL* formula $\forall[P\,\mathbf{U}\,\exists\mathbf{F}Q]$ is also a CTL formula, but $\forall[P\,\mathbf{U}\,\mathbf{F}Q]$ is not, because the $\mathbf{F}$ is not immediately governed by a quantifier. Hence extensions also cover the traditional temporal logics described in Chapter 11 of this Handbook — that is, literally logics of time — as advocated by Manna and Pnueli and others. In this view, time is a linear sequence of instants, corresponding to the states of just one execution path through the program. One can define a logic on paths which has operators $\bigcirc\phi$ 'in the next instant (on this path) ϕ is true', and $\phi\,\mathbf{U}\,\psi$ 'ϕ holds until ψ holds (on this path)'; and then a system satisfies a formula if all execution paths satisfy the formula — in CTL* terms, the specification is a path formula with a single outermost universal quantifier. One can also extend PDL with temporal operators, as in process logic.

There are extensions of all these logics to cover issues such as time and probability. The introduction of such real-valued quantities poses a number of problems, and such logics are still under active development.

2.3 The small model property

A general result about many modal logics is that they have the small model property; that is if a formula is satisfiable, then it is satisfiable by a model of some bounded size. Provided that model-checking is decidable, which is the case for almost all temporal

logics, this immediately gives decidability of satisfiability for the logic, as one need simply check every transition system up to the size bound.

The technique used to establish the small model property for PDL (and therefore HML) is a classical technique in modal logic, that of *filtration*. Let s be a state, satisfying property ϕ, in a possibly infinite transition system $\mathfrak{T}$. Let Γ be the set of all subformulas of ϕ and their negations: in the case of PDL one also counts $\langle\alpha\rangle\psi$ and $\langle\beta\rangle\psi$ as subformulas of $\langle\alpha \cup \beta\rangle\psi$, $\langle\beta\rangle\psi$ as a subformula of $\langle\alpha;\beta\rangle\psi$ and $\langle\alpha;\alpha^*\rangle\psi$, and $\langle\alpha\rangle\psi$ as subformulas of $\langle\alpha^*\rangle\psi$. The size of Γ is proportional to $|\phi|$ (the length of ϕ). One then filters $\mathfrak{T}$ through Γ by defining an equivalence relation on the states of $\mathfrak{T}$, $s \equiv t$ iff $\forall\psi \in \Gamma . s \models \psi \Leftrightarrow t \models \psi$. We define the filtered model to have states $\mathfrak{T}/\equiv$ and with atomic action relations given by $[s] \xrightarrow{a} [t]$ iff $\exists s' \in [s], t' \in [t]. s' \xrightarrow{a} t'$. The number of equivalence classes is at most $2^{|\Gamma|}$ and so is $O(2^{|\phi|})$. The rest of the proof shows that the filtered model is indeed a model, in that $[s] \models \psi$ iff $s \models \psi$ for $\psi \in \Gamma$. For PDL the only case requiring comment is the case $\langle\alpha^*\rangle\psi$, which proceeds by an induction on the length of the witnessing sequence of α's. Consequently if ϕ is a satisfiable PDL formula, then it has a model with size $O(2^{|\phi|})$, and in fact $2^{|\phi|}$ suffices — see [25] for full details.

In order to obtain an upper bound for satisfiability from the small model property, we also need to know the complexity of model-checking, that is, determining whether $s \models \phi$. It is straightforward to define an inductive procedure for this, which is polynomial in the size of the formula and of the system. For example, to determine the truth of $\langle\alpha^*\rangle\phi$, one computes the *-closure of the α relation, and then checks for an α^*-successor satisfying ϕ. These results give an NTIME(c^n) (where c is a constant and n the formula size) upper bound for the satisfiability problem. By a reduction to alternating Turing machines, [25] also gave a lower bound of DTIME($c^{n/\lg n}$). A closer to optimal technique for satisfiability due to Pratt uses tableaux [58].

Although CTL, CTL* and $L\mu$ all have the finite model property, the filtration technique does not apply. If one filters $\mathfrak{T}$ through a finite set Γ containing $\forall\mathbf{F}Q$ unintended loops may be added. For instance if $\mathfrak{T}$ is $s_i \xrightarrow{a} s_{i+1}$ for $1 \leq i < n$ and Q is only true at state s_n then $s_i \models \forall\mathbf{F}Q$ for each i. But when n is large enough the filtered model will have at least one transition $[s_j] \xrightarrow{a} [s_i]$ when $i \leq j < n$, with the consequence that $[s_i] \not\models \forall\mathbf{F}Q$. The initial approach to showing the finite model property utilized semantic tableaux where one explicitly builds a model for a satisfiable formula which has small size. But such a technique is very particular, and more sophisticated methods based on automata are used for optimal results, as we shall mention later.

3 SYNTAX AND SEMANTICS OF MODAL MU-CALCULUS

The defining feature of mu-calculi is the use of fixpoint operators. The use of fixpoint operators in program logics goes back at least to De Bakker, Park and Scott [56]. However, their use in modal logics of programs dates from work of Pratt, Emerson and Clarke and Kozen. Pratt's version [59] used a fixpoint operator like the minimization operator of recursion theory; although this is only superficially different, it seems to have dissuaded people from using the logic in that form. Emerson and Clarke added fixed points to a temporal logic to capture fairness and other correctness properties [21]. Kozen's [35] paper introduced $L\mu$ as we use it today, and established a number of basic results.

Fixpoint logics are traditionally considered hard to understand. Furthermore, their

semantics requires a familiarity with material that, although not difficult, is often omitted from undergraduate mathematics or logic programmes. Whether for practical purposes, or to guide oneself through the formal proofs, it is therefore worthwhile to spend a little time on discussing an intuitive understanding of $L\mu$ before going on to the definitions.

3.1 Fixpoints as recursion

Suppose that $\mathfrak{S}$ is the state space of some system. For example $\mathfrak{S}$ could be the set of all processes reachable by arbitrary length sequences of transitions from some initial process. One way to provide semantics of a state-based modal logic is to map formulae ϕ to sets of states, that is to elements of $\wp\mathfrak{S}$. For any formula ϕ this mapping is given by $\|\phi\|$[1]. The idea is that this mapping tells us at which states each formula holds. If we allow our logic to contain variables with interpretations ranging over $\wp\mathfrak{S}$, then we can view the semantics of a formula with a free variable, $\phi(Z)$, as a function $f \colon \wp\mathfrak{S} \to \wp\mathfrak{S}$. If $f(S) \subseteq f(S')$ whenever $S \subseteq S' \subseteq \mathfrak{S}$ then f is *monotonic*. If $f(S) = S$ then S is a *fixed point* of f (as repeated application of f leaves S unchanged). If we take the usual lattice structure on $\wp\mathfrak{S}$, given by set inclusion, and if f is a monotonic function, then by the Knaster–Tarski theorem we know that f has fixed points, and indeed has a unique maximal and a unique minimal fixed point. The maximal fixed-point is the union of *post-fixed points*, $\bigcup\{S \subset \mathfrak{S} \mid S \subseteq f(S)\}$, and the minimal fixed-point is the intersection of *pre-fixed points*, $\bigcap\{S \subset \mathfrak{S} \mid f(S) \subseteq S\}$. So we could extend our basic logic with a minimal fixpoint operator μ, so that $\mu Z.\phi(Z)$ is a formula whose semantics is the least fixed point of f; and similarly a maximal fixpoint operator ν, so that $\nu Z.\phi(Z)$ is a formula whose semantics is the greatest fixed point of f (when the semantics of $\phi(Z)$ is monotonic).

A good reason to do this is that it provides a semantics for *recursion*, and adding recursion to the usual modal logics provides a neat way of expressing all the usual operators of temporal logics. For example, consider the CTL formula $\forall\mathbf{G}\phi$, 'always ϕ'. Another way of expressing this is to say that it is a property X such that if X is true, then ϕ is true, and wherever we go next, X remains true; so X satisfies the modal implicational equation

$$X \Rightarrow \phi \wedge [-]X.$$

where $[-]X$ means that X is true at every immediate successor (see subsection 3.3). A solution to this equation is precisely a post-fixed point of the formula $\phi \wedge [-]X$. But which solution of the possibly many is appropriate? The only canonical post-fixed point is the largest, and this also makes sense, since if a state satisfies *some* solution, then it surely satisfies $\forall\mathbf{G}\phi$. Hence the meaning of the formula is the largest post-fixed point, which by standard theory is exactly the largest fixed point, $\nu X.\phi \wedge [-]Z$.

On the other hand, consider the CTL property $\exists\mathbf{F}\phi$, 'there exists a path on which ϕ eventually holds'. We could write this recursively as 'Y holds if either ϕ holds now, or there's some successor on which Y is true':

$$Y \Leftarrow \phi \vee \langle - \rangle Y.$$

Here we have a pre-fixed point of $\phi \vee \langle - \rangle Y$; the only canonical such is the least, and if a state satisfies $\exists\mathbf{F}\phi$, then it surely satisfies any solution Y' of the equation. Hence we want the least pre-fixed point, which is also the least fixed point, $\mu Y.\phi \vee \langle - \rangle Y$.

[1]The mapping can be either given directly (inductively) or indirectly as the set $\{s \in \mathfrak{S} : s \models \Phi\}$.

Finally, we observe that since we want the fixed points, we may replace the implications by equalities in the modal equations above, and get the same answers. It is therefore usual to cast modal fixpoint logics in terms of equations, rather than of implications.

3.2 Approximating fixpoints and μ as 'finitely'

The other key idea is that of approximants and unfolding. The standard theory tells us that if f is a monotonic function on a lattice, we can construct the least fixed point of f by applying f repeatedly on the bottom element $\perp$ of the lattice to form an increasing chain, whose limit is the fixed point. The length of the iteration is in general transfinite, but is bounded at worst by the cardinal after cardinality of the lattice, and in the special case of a powerset lattice $\wp\mathfrak{S}$, by the cardinal after the cardinality of $\mathfrak{S}$. So if f is monotonic on $\wp\mathfrak{S}$, we have the increasing chain $\emptyset \subseteq f(\emptyset) \subseteq f^2(\emptyset) \subseteq \ldots \subseteq f^\alpha(\emptyset) \ldots$ and the least fixed point is the limit of this chain

$$\mu f = \bigcup_{\alpha<\kappa} f^\alpha(\emptyset)$$

and similarly as there is the anti-chain, $\mathfrak{S} \supseteq f(\mathfrak{S}) \supseteq f^2(\mathfrak{S}) \supseteq \ldots \supseteq f^\alpha(\mathfrak{S}) \ldots$,

$$\nu f = \bigcap_{\alpha<\kappa} f^\alpha(\mathfrak{S})$$

—or in terms of a infinitary logic, $\mu Z.\phi(Z) = \bigvee_{\alpha<\kappa} \phi^\alpha(\perp)$—where κ is at worst $|\mathfrak{S}|+1$ for finite $\mathfrak{S}$, or $\aleph_1$ for countable $\mathfrak{S}$ (and $\nu Z.\phi(Z) = \bigwedge_{\alpha<\kappa} \phi^\alpha(\top)$). So for a minimal fixpoint $\mu Z.\phi(Z)$, if a state s satisfies the fixpoint, it satisfies some approximant, say for convenience the $\beta+1$th so that $s \models \phi^{\beta+1}(\perp)$. Now if we *unfold* this formula once, we get $s \models \phi(\phi^\beta(\perp))$. That is, the fact that s satisfies the fixpoint depends, via ϕ, on the fact that other states satisfy the fixpoint *at smaller approximants than s does.* So if one follows a chain of dependencies, the chain terminates. This is the strict meaning behind the slogan "μ means 'finite looping'", which, with a little refinement, is sufficient to understand $L\mu$.

On the other hand, for a maximal fixpoint $\nu Z.\phi(Z)$, there is no such decreasing chain: $s \models \nu Z.\phi(Z)$ iff $s \models \phi(\nu Z.\phi(Z))$, and we may loop for ever, as in the process $P \stackrel{\text{def}}{=} a.P$, which repeatedly does an a action, and so satisfies $\nu Z.\langle a\rangle Z$. (However, if a state *fails* a maximal fixpoint, then there is a descending chain of failures.) Instead, we have the principle of fixpoint induction: if by assuming that a set $S \models Z$, we can show that $S \models \phi(Z)$, then we have shown that $S \models \nu Z.\phi$ (compare the recursive formulation of $\forall\mathbf{G}\phi$ in the previous section).

So in summary, one may understand fixpoints by the slogan 'ν means looping, and μ means finite looping'. This slogan provides an alternative means of explaining why a minimal fixpoint is required in the translation of $\exists\mathbf{F}\phi$. This formula means that there is a path on which ϕ eventually holds: that is, on the chosen path, ϕ holds within finite time. Hence the 'equation' $Y = \phi \vee \langle -\rangle Y$ must only be applied a finite number of times, and so by the slogan we should use a minimal fixpoint.

In the case of formulae with alternating fixpoints (which we shall examine a little later), the slogan remains valid, but requires a little more care in application. It is essential to almost all proofs about $L\mu$: the notion of 'well-founded premodel' with which Streett

and Emerson [64] proved the finite model property, is an example of the slogan; so are the tableau model-checking approaches of Stirling and Walker [62], and Bradfield and Stirling [12].

3.3 Syntax of $L\mu$

Let Var be an (infinite) set of *variable names*, typically indicated by $Z, Y, \ldots$; let Prop be a set of *atomic propositions*, typically indicated by $P, Q, \ldots$; and let $\mathcal{L}$ be a set of *labels*, typically indicated by $a, b, \ldots$. The set of $L\mu$ formulae (with respect to $\mathrm{Var}, \mathrm{Prop}, \mathcal{L}$) is defined in parsimonious form as follows:

- P is a formula.
- Z is a formula.
- If ϕ_1 and ϕ_2 are formulae, so is $\phi_1 \wedge \phi_2$.
- If ϕ is a formula, so is $[a]\phi$.
- If ϕ is a formula, so is $\neg\phi$.
- If ϕ is a formula, then $\nu Z.\phi$ is a formula, provided that every free occurrence of Z in ϕ occurs positively, i.e. within the scope of an even number of negations. (The notions of free and bound variables are as usual, where ν is the only binding operator.)

If a formula is written as $\phi(Z)$, it is to be understood that the subsequent writing of $\phi(\psi)$ means ϕ with ψ substituted for all free occurrences of Z. There is no suggestion that Z is the only free variable of ϕ.

The positivity requirement on the fixpoint operator is a syntactic means of ensuring that $\phi(Z)$ denotes a functional monotonic in Z, and so has unique minimal and maximal fixpoint. It is usually more convenient to introduce derived operators defined by de Morgan duality, and work in positive form:

- $\phi_1 \vee \phi_2$ means $\neg(\neg\phi_1 \wedge \neg\phi_2)$.
- $\langle a\rangle\phi$ means $\neg[a]\neg\phi$.
- $\mu Z.\phi(Z)$ means $\neg\nu Z.\neg\phi(\neg Z)$.

Note the triple use of negation in μ, which is required to maintain the positivity. A formula is said to be in *positive form* if it is written with the derived operators so that $\neg$ only occurs applied to atomic propositions. It is in *positive normal form* if in addition all bound variables are distinct (and different from free variables). Any formula can be put into positive normal form by use of de Morgan laws and α-conversion. So we shall often assume positive normal form, and when doing structural induction on formulae will often take the derived operators as primitives.

For the concrete syntax, we shall assume that modal operators have higher precedence than boolean, and that fixpoint operators have lowest precedence, so that the scope of a fixpoint extends as far to the right as possible.

There are a few extensions to the syntax which are convenient in presenting examples, and in practice. The most useful is to allow modalities to refer not just to single actions, but to sets of actions. The most useful set is ‘all actions except a’. So:

- $s \models [K]\phi$ iff $\forall a \in K. s \models [a]\phi$, and $[a, b, \ldots]\phi$ is short for $[\{a, b, \ldots\}]\phi$.
- $[-K]\phi$ means $[\mathcal{L} - K]\phi$, and set braces may be omitted.

Thus $[-]\phi$ means just $[\mathcal{L}]\phi$. [2]

3.4 Semantics of $L\mu$

An *$L\mu$ structure* $\mathfrak{T}$ (over Prop, $\mathcal{L}$) is a labelled transition system, namely a set $\mathfrak{S}$ of states and a transition relation $\rightarrow \subseteq \mathfrak{S} \times \mathcal{L} \times \mathfrak{S}$ (as usual we write $s \xrightarrow{a} t$), together with an interpretation $\mathfrak{V}_{\mathrm{Prop}}\colon \mathrm{Prop} \rightarrow \wp\mathfrak{S}$ for the atomic propositions.

Given a structure $\mathfrak{T}$ and an interpretation $\mathfrak{V}\colon \mathrm{Var} \rightarrow \wp\mathfrak{S}$ of the variables, the set $\|\phi\|^{\mathfrak{T}}_{\mathfrak{V}}$ of states satisfying a formula ϕ is defined as follows:

$$
\begin{aligned}
\|P\|^{\mathfrak{T}}_{\mathfrak{V}} &= \mathfrak{V}_{\mathrm{Prop}}(P) \\
\|Z\|^{\mathfrak{T}}_{\mathfrak{V}} &= \mathfrak{V}(Z) \\
\|\neg\phi\|^{\mathfrak{T}}_{\mathfrak{V}} &= \mathfrak{S} - \|\phi\|^{\mathfrak{T}}_{\mathfrak{V}} \\
\|\phi_1 \wedge \phi_2\|^{\mathfrak{T}}_{\mathfrak{V}} &= \|\phi_1\|^{\mathfrak{T}}_{\mathfrak{V}} \cap \|\phi_2\|^{\mathfrak{T}}_{\mathfrak{V}} \\
\|[a]\phi\|^{\mathfrak{T}}_{\mathfrak{V}} &= \{\, s \mid \forall t. s \xrightarrow{a} t \Rightarrow t \in \|\phi\|^{\mathfrak{T}}_{\mathfrak{V}} \,\} \\
\|\nu Z.\phi\|^{\mathfrak{T}}_{\mathfrak{V}} &= \bigcup \left\{ S \subseteq \mathfrak{S} \mid S \subseteq \|\phi\|^{\mathfrak{T}}_{\mathfrak{V}[Z:=S]} \right\}
\end{aligned}
$$

where $\mathfrak{V}[Z := S]$ is the valuation which maps Z to S and otherwise agrees with $\mathfrak{V}$. If we are working in positive normal form, we may add definitions for the derived operators by duality:

$$
\begin{aligned}
\|\phi_1 \vee \phi_2\|^{\mathfrak{T}}_{\mathfrak{V}} &= \|\phi_1\|^{\mathfrak{T}}_{\mathfrak{V}} \cup \|\phi_2\|^{\mathfrak{T}}_{\mathfrak{V}} \\
\|\langle a\rangle\phi\|^{\mathfrak{T}}_{\mathfrak{V}} &= \{\, s \mid \exists t. s \xrightarrow{a} t \wedge t \in \|\phi\|^{\mathfrak{T}}_{\mathfrak{V}} \,\} \\
\|\mu Z.\phi\|^{\mathfrak{T}}_{\mathfrak{V}} &= \bigcap \left\{ S \subseteq \mathfrak{S} \mid S \supseteq \|\phi\|^{\mathfrak{T}}_{\mathfrak{V}[Z:=S]} \right\}
\end{aligned}
$$

We drop $^{\mathfrak{T}}$ and $_{\mathfrak{V}}$ whenever possible; and write $s \models \phi$ for $s \in \|\phi\|$.

We have discussed informally the importance of approximants; let us now define them. If $\mu Z.\phi(Z)$ is a formula, then for an ordinal α, let $\mu Z^{\alpha}.\phi$ and $\mu Z^{<\alpha}.\phi$ be formulae, with semantics given, with simultaneous induction on α, by:

$$
\begin{aligned}
\|\mu Z^{<\alpha}.\phi\|^{\mathfrak{T}}_{\mathfrak{V}} &= \bigcup_{\beta<\alpha} \|\mu Z^{\beta}.\phi\|^{\mathfrak{T}}_{\mathfrak{V}} \\
\|\mu Z^{\alpha}.\phi\|^{\mathfrak{T}}_{\mathfrak{V}} &= \|\phi\|^{\mathfrak{T}}_{\mathfrak{V}[Z:=\|\mu Z^{<\alpha}.\phi\|^{\mathfrak{T}}_{\mathfrak{V}}]}
\end{aligned}
$$

The approximants of a maximal fixpoint are defined dually:

$$
\begin{aligned}
\|\nu Z^{<\alpha}.\phi\|^{\mathfrak{T}}_{\mathfrak{V}} &= \bigcap_{\beta<\alpha} \|\nu Z^{\beta}.\phi\|^{\mathfrak{T}}_{\mathfrak{V}} \\
\|\nu Z^{\alpha}.\phi\|^{\mathfrak{T}}_{\mathfrak{V}} &= \|\phi\|^{\mathfrak{T}}_{\mathfrak{V}[Z:=\|\nu Z^{<\alpha}.\phi\|^{\mathfrak{T}}_{\mathfrak{V}}]}
\end{aligned}
$$

[2] Beware that many authors use '$[\,]\phi$' to mean '$[\mathcal{L}]\phi$', rather than the (vacuous) '$[\emptyset]\phi$' that it means in our notation.

Note that $\mu Z^{<0}.\phi \Leftrightarrow \bot$ and $\nu Z^{<0}.\phi \Leftrightarrow \top$. By abuse of notation, we write Z^α or ϕ^α to mean ${}^{\mu}_{\nu}Z^\alpha.\,\phi$; of course this only makes sense when one knows which fixpoint and variable is meant.

We should remark here that most literature on $L\mu$ uses a slightly different definition, putting $\mu Z^0.\phi = \bot$, $\mu Z^{\alpha+1}.\phi = \phi(\mu Z^\alpha.\phi)$, and $\mu Z^\lambda.\phi = \bigcup_{\beta<\lambda} \mu Z^\beta.\phi$ for limit λ—which in effect is writing α for our $<\alpha$. That notation is taken from set theory; its advantage is that a limit approximant is the limit of approximants. Our notation is taken from more recent set theoretic practice; its advantages are that it sometimes reduces the number of trivial case distinctions in inductive proofs. However, the difference is not significant.

Sometimes, we are interested in *rooted* structures $(\mathfrak{T}, s_0, \mathfrak{V}_{\text{Prop}})$ for $L\mu$ formulae that have a designated initial state s_0: ϕ is true of such a structure if $s_0 \models \phi$. We can, therefore, examine the set of all rooted structures where ϕ is true which allows comparison between $L\mu$ and other notations for classifying structures.

3.5 Examples

We have seen, both informally and in the formal semantics, the meaning of the fixpoint operators, and we have seen some simple examples of $L\mu$ translating CTL. We now consider some examples of $L\mu$ formulae in their own right, which express properties one might meet in practice.

There is a well-known ‘classification’ [42] of basic properties into safety and liveness. In terms of $L\mu$, it is not unreasonable to say that μ is liveness and ν is safety. Consider first simple ν formulae. For example:

$$\nu Z.P \wedge [a]Z$$

is a relativized ‘always’ formula: ‘P is true along every a-path’. Slightly more complex is the relativized ‘while’ formula

$$\nu Z.Q \vee (P \wedge [a]Z)$$

‘on every a-path, P holds while Q fails’. Both formulae can be understood directly via the fixpoint construction, or via the idea of ‘ν as looping’: for example the second formula is true if either Q holds, or if P holds and wherever we go next (via a), the formula is true, and ..., and because the fixpoint is maximal, we can repeat forever. So in particular, if P is always true, and Q never holds, the formula is true.

μ formulae, in contrast, require something to happen, and thus are liveness properties. For example

$$\mu Z.P \vee [a]Z$$

is ‘on all infinite length a-paths, P eventually holds’; and

$$\mu Z.Q \vee (P \wedge \langle a\rangle Z)$$

is ‘on some a-path, P holds until Q holds (and Q *does* eventually hold)’. Again, these can be understood by ‘μ as finite looping’: in the second case, we are no longer allowed to repeat the unfolding forever, so we must eventually ‘bottom out’ in the Q disjunct.

This level of complexity suffices to translate CTL, since we have $\mu Z.Q \vee (P \wedge \langle -\rangle Z)$ as a translation of $\exists[P\,\mathbf{U}\,Q]$, and $\mu Z.Q \vee (P \wedge [-]Z \wedge \langle -\rangle\top)$ as a translation of $\forall[P\,\mathbf{U}\,Q]$

(the conjunct $\langle - \rangle \top$ ensures that Q is actually reached, since $[-]Z$ is true at deadlock states); and obviously we can nest formulae inside one another, such as

$$\nu Z.(\mu Y.P \vee \langle - \rangle Y) \wedge [-]Z$$

'it is always possible that P will hold', or $\forall \mathbf{G}(\exists \mathbf{F} P)$. Equally obviously, we can write formulae with no CTL translation, such as

$$\mu Z.[a]\bot \vee \langle a \rangle \langle a \rangle Z$$

which asserts the existence of a maximal a-path of even length; a formula which is, incidentally, expressible in PDL. This is, however, a fairly simple extension; much more interesting is the power one gets from mixing fixpoints that depend on one another. Consider the formula

$$\mu Y.\nu Z.(P \wedge [a]Y) \vee (\neg P \wedge [a]Z).$$

This formula usually gives pause for thought, but it has a simple meaning, which can be seen by using the slogans. $\mu Y\ldots$ is true if $\nu Z\ldots$ is true if $(P \wedge [a]Y) \vee (\neg P \wedge [a]Z)$, which is true if either P holds and at the next (a)-states we loop back to $\mu Y\ldots$, or P fails, and at the next states we loop back to $\nu Z\ldots$. By the slogan 'μ means finitely', we can only loop through $\mu Y\ldots$ finitely many times on any path, and hence P is true only finitely often on any path.

We shall see in a later section that this so-called alternation of fixpoint operators does indeed give ever more expressive power as the number of alternations increases. It also appears to increase the complexity of model-checking: all known algorithms are exponential in the alternation, but whether this is necessarily the case is the main remaining open problem about $L\mu$.

3.6 Fixpoint regeneration and the 'fundamental semantic theorem'

In the informal description of the meaning of fixpoints, we used the idea of the dependency of s at ϕ on t at ψ. We now make this precise. Assume a structure $\mathfrak{T}$, and a formula ϕ. Suppose that we annotate the states with sets of subformulae, such that the sets are locally consistent: that is, s is annotated with a conjunction iff it is annotated with both conjuncts; s is annotated with a disjunction iff it is annotated with at least one disjunct; if s is annotated with $[a]\psi$ (resp. $\langle a \rangle \psi$), then each (resp. at least one) a-successor is annotated with ψ; if s is annotated with a fixpoint or fixpoint variable, it is annotated with the body of the fixpoint. We call such an annotated structure a *quasi-model*.

A *choice function* f is a function which for every disjunctive subformula $\psi_1 \vee \psi_2$ and every state annotated with $\psi_1 \vee \psi_2$ chooses one disjunct $f(s, \psi_1 \vee \psi_2)$; and for every subformula $\langle a \rangle \psi$ and every state s annotated with $\langle a \rangle \psi$ chooses one a-successor $t = f(s, \langle a \rangle \psi)$ annotated with ψ.

A *pre-model* is a quasi-model equipped with a choice function.

Given a pre-model with choice function f, the *dependencies* of a state s that satisfies a subformula ψ are defined thus: $s@\psi_1 \wedge \psi_2 \succ s@\psi_i$ for $i = 1, 2$; $s@[a]\psi \succ t@\psi$ for every t such that $s \xrightarrow{a} t$; $s@\psi_1 \vee \psi_2 \succ s@f(s, \psi_1 \vee \psi_2)$; $s@\langle a \rangle \psi \succ f(s, \langle a \rangle \psi)@\psi$; $s@{}^{\mu}_{\nu}Z.\,\psi \succ s@\psi$; $s@Z \succ s@\psi$ where Z is bound by ${}^{\mu}_{\nu}Z.\,\psi$. A *trail* is a maximal chain of dependencies.

If every trail has the property that the highest (i.e. with the outermost binding fixpoint) variable occurring infinitely often is a ν-variable, the pre-model is *well-founded*.

(Equivalently: in any trail, a μ-variable can only occur finitely often unless a higher variable is encountered.)

The fundamental theorem on the semantics of $L\mu$ can now be stated:

THEOREM 1. A well-founded pre-model is a model: in a well-founded pre-model, if s is annotated with ψ, then indeed $s \models \psi$.

The theorem in this form is due to Streett and Emerson in [64], from which the term 'well-founded pre-model' is taken. Stirling and Walker [63] presented a tableau system for model-checking on finite structures, and the soundness theorem for that system is essentially a finite version of the fundamental theorem using a more relaxed notion of choice; the later infinite-state version of [12, 7] is the fundamental theorem, again with a slight relaxation on choice.

A converse is also true:

THEOREM 2. If in some structure $s \models \phi$, then there is a *locally consistent* annotation of the structure and a choice function which make the structure a well-founded pre-model.

The fundamental theorem, in its various guises, is the precise statement of the slogan 'μ means finite looping'. To explain why it is true, and to define the term 'locally consistent', we need to make a finer analysis of approximants.

Assume a structure $\mathfrak{T}$, valuation $\mathfrak{V}$, and formula ϕ in positive normal form. Let $Y_1, \ldots, Y_n$ be the μ-variables of ϕ, in an order compatible with formula inclusion: that is, if $\mu Y_j.\psi_j$ is a subformula of $\mu Y_i.\psi_i$, then $i \leq j$. If Y_i is some inner fixpoint, then its denotation depends on the meaning of the fixpoints enclosing it: for example, in the formula $\mu Y_1.\langle a\rangle \mu Y_2.(P \vee Y_1) \vee \langle b\rangle Y_2$, to calculate the inner fixpoint μY_2 we need to know the denotation of Y_1. We may ask: what is the least approximant of Y_1 that could be plugged in to make the formula true? Having fixed that, we can then ask what approximant of Y_2 is required. This idea is the notion of *signature*. A signature is a sequence $\sigma = \alpha_1, \ldots, \alpha_n$ of ordinals, such that the i least fixpoint will be interpreted by its α_ith approximant (calculated relative to the outer approximants).

The definition and use of signatures inevitably involves some slightly irritating book-keeping, and they appear in several forms in the literature. In [64], the Fischer–Ladner closure of ϕ was used, rather than the set of subformulae. The closure is defined by starting with ϕ and closing under the operations of taking the immediate components of formulae with boolean or modal top-level connectives, together with the rule that if ${}^{\mu}_{\nu}Z.\,\psi(Z) \in \mathrm{cl}(\phi)$, then $\psi({}^{\mu}_{\nu}Z.\,\psi) \in \mathrm{cl}(\phi)$. The signatures were defined by syntactically unfolding fixpoints, rather than by semantic approximants. In [63] and following work, a notion of *constant* was used, which allows some of the book-keeping to be moved into the logic. Although all the notions and proofs using them are interconvertible, the 'constant' variant is perhaps easier to follow, and has the advantage that it adapts easily to the modal equation system presentation of $L\mu$, which we shall see below. Indeed, it arises more naturally from that system.

Add to the language a countable set of *constants* $U, V, \ldots$. Constants will be defined to stand for maximal fixpoints or approximants of minimal fixpoints. Specifically, given a formula ϕ, let $Y_1, \ldots, Y_n$ be the μ-variables as above, let $Z_1, \ldots, Z_m$ be the ν-variables, let $\sigma = \alpha_1, \ldots, \alpha_n$ be a signature, and let $U_1, \ldots, U_n, V_1, \ldots, V_m$ be constants, which will be associated with the corresponding variables. They are given semantics thus: if Y_i is bound by $\mu Y_i.\psi_i$, then $\|U_i\|_\sigma$ is $\|\mu Y_i^{\alpha_i}.\psi_i'\|_\sigma$, where ψ_i' is obtained from ψ_i by substituting the corresponding constants for the free fixpoint variables of $\mu Y_i.\psi_i$. If Z_i is bound by

$\nu Z_i.\psi_i$, its semantics is $\|\nu Z_i.\psi_i'\|_\sigma$. Given an arbitrary subformula ψ of ϕ, we say a state s satisfies ψ with signature σ, written $s \models_\sigma \psi$, if $s \in \|\psi'\|_\sigma$, where ψ' is ψ with its free fixpoint variables substituted by the corresponding constants.

Order signatures lexicographically. Now, given a pre-model for ϕ, extend the annotations so that each subformula at s is accompanied by a signature – write $s@\psi[\sigma]$. Such an extended annotation is said to be locally consistent if the signature is unchanged or decreases by passing through boolean, modal, or ν-variable dependencies, and when passing through $s@Y_i$ it strictly decreases in the ith component and is unchanged in the $1, \ldots, i-1$'th components.

It can now be shown, by a slightly delicate but not too difficult induction on ψ and σ, that if $s@\psi[\sigma]$, then $s \models_\sigma \psi$. The proof proceeds by contradiction: suppose that $s@\psi[\sigma]$ and $s \not\models_\sigma \psi$. If ψ is $\psi_1 \vee \psi_2$ ($\psi_1 \wedge \psi_2$) then for some $i \in \{1,2\}$, $s@\psi_1[\sigma]$ and $s \not\models_\sigma \psi_i$. If ψ is $[a]\psi'$ ($\langle a\rangle\psi'$) then for some s', $s \xrightarrow{a} s'$, $s'@\psi'[\sigma]$ and $s' \not\models_\sigma \psi'$. If ψ is a least fixpoint variable Y_i, then we pass through the unfolding rule to $s@\psi_i[\sigma']$ where $\sigma' < \sigma$ and $s \not\models_{\sigma'} \psi_i$. (Hence we can only pass through least fixpoints finitely often.) The key case is when ψ is a greatest fixpoint variable Z_i. In this case, the hypothesis is that $s@Z_i[\sigma]$ and $s \not\models_\sigma Z_i$. Then s fails some approximant Z_i^β (and $s@Z_i^\beta[\sigma]$); and then passing through the unfolding rule gives s fails $\psi_i^{\beta'}$ for $\beta' < \beta$ (and $s@\psi_i^{\beta'}[\sigma]$). Continuing to chase the falsehood down the pre-model, we eventually arrive at a state failing the zero'th approximant of a greatest fixpoint formula, which is impossible.

Furthermore, given a well-founded pre-model, one can construct a locally consistent signature annotation—essentially, the Y_i component of σ in $s@\psi[\sigma]$ is the maximum 'number' (in the transfinite sense) of Y_i occurrences without meeting a higher variable in trails from $s@\psi$, and so on; the well-foundedness of the pre-model guarantees that this is well-defined. This gives the fundamental theorem.

The converse is quite easy: given a model, annotate the states by the subformulae they satisfy; for $s@\psi$ assign the least σ such that $s \models_\sigma \psi$; and choose a choice function that always chooses the successor with least signature. It is easy to show that this is a well-founded pre-model and signature assignment.

3.7 Modal equation systems

The presentation of $L\mu$ so far is a traditional logical formulation. However, in several circumstances it can be useful to think in terms of systems of recursive equations for the fixpoint variables, as follows.

A *modal equation system* comprises a sequence $B_0; \ldots; B_n$ of *blocks*; each B_i may be a *μ-block* (we write B_i^μ) or a *ν-block* (we write B_i^ν). Each block $B_i^{\mu/\nu}$ is a sequence of equations $X_{i0} \stackrel{\mu/\nu}{=} \phi_{i0}, \ldots, X_{ik_i} \stackrel{\mu/\nu}{=} \phi_{ik_i}$, where each ϕ_{ij} is a modal formula which may contain any of the variables $X_{i'j'}$ positively.

Thus each block B_i defines a functional on vectors $(S_{i0}, \ldots, S_{ik_i}) \in (\wp\mathfrak{S})^{k_i}$. This functional is relative to valuations of the variables in earlier blocks, and refers to the solutions of later blocks. If B_i^μ, then take the least fixpoint (in the componentwise ordering) of this functional, and if B_i^ν, take the greatest. Conventionally, the solution of the entire equation system is taken to be the solution for the first variable X_{00}.

There is an obvious transformations from $L\mu$ to modal equation systems: for example,

$\mu X.P \vee \nu Y.[a]Y \wedge [b]X$ translates to

$$X_{00} \stackrel{\mu}{=} P \vee X_{10} \quad ; \quad X_{10} \stackrel{\nu}{=} [a]X_{10} \wedge [b]X_{00}.$$

Similarly, there is a reasonably obvious reverse transformation: for example, the equation system

$$X_{00} \stackrel{\mu}{=} \langle a \rangle X_{10} \vee [b]X_{10} \quad ; \quad X_{10} \stackrel{\nu}{=} P \wedge [a](X_{00} \vee X_{10})$$

translates to $\mu X.\langle a \rangle(\nu Y.P \wedge [a](X \vee Y)) \vee [b](\nu Y.P \wedge [a](X \vee Y))$. These translations, known from finite model theory, show that modal equation systems and $L\mu$ are equi-expressive. Note that in the second example, the formula duplicates the second equation: by extending such examples, one can see that the translation from equation systems to formulae may introduce an exponential blow-up. However, this blow-up results in formulae with many identical sub-formulae, which can in any case be optimized away during model-checking, and in general problems in modal equation systems are of the same complexity as in $L\mu$.

A block in a modal equation system is to be understood as a *simultaneous* fixpoint. $L\mu$ could be directly presented with simultaneous fixed points: for instance, $s \models \mu Z_1 \ldots Z_n.(\phi_1, \ldots, \phi_n)$ iff $s \in S_1$ where $(S_1, \ldots, S_n) = \bigcap \{(S'_1, \ldots, S'_n) \mid S'_j \supseteq \|\phi_j\|^{\mathfrak{T}}_{\mathfrak{V}[Z_1:=S'_1,\ldots,Z_n:=S'_n]}\}$.

One of the main applications of modal equation systems is in the development of fast model-checking algorithms: modal equation systems can be easily translated to *boolean equation systems* (defined as above, but with boolean variables and just propositional equations) by having one boolean variable for each (modal variable, state) pair. Then graph-theoretic or matrix-theoretic techniques can be employed to solve the boolean equation systems. For more on this topic, see [46].

4 EXPRESSIVE POWER

As we noted earlier in this article, there are many temporal logics used in practice, some of which are also historical precursors to $L\mu$. We said that most of them could be seen as fragments of $L\mu$. In this section we consider questions of expressivity and related topics, and start by showing how a number of other logics can be translated into $L\mu$.

4.1 CTL and friends as fragments of $L\mu$

PDL can be easily translated into $L\mu$ by unpacking the modal operators $\langle \alpha \rangle$: $\langle \alpha \cup \beta \rangle \psi = \langle \alpha \rangle \psi \vee \langle \beta \rangle \psi$, $\langle \alpha ; \beta \rangle \psi = \langle \alpha \rangle \langle \beta \rangle \psi$ and $\langle \alpha^* \rangle \psi = \mu Z.\psi \vee \langle \alpha \rangle Z$. The logic CTL is one of the simplest temporal logics, and its translation is also simple. Recall the syntax and semantics of CTL from 2.2. The two basic operators are $\forall[\phi \mathbf{U} \psi]$ and $\exists[\phi \mathbf{U} \psi]$. Assuming that there are no deadlocked states, these can be simply translated as:

$$\mu Z.\psi \vee [-]\phi \qquad \text{and} \qquad \mu Z.\psi \vee \langle - \rangle \phi$$

with the proof of the equivalence being a straightforward application of the semantics. For both PDL and CTL, only a fragment of $L\mu$ is necessary where there is no essential alternation of fixpoints (as described in 7).

A much less trivial case is the logic CTL*. CTL* is the logic obtained by removing the syntactic constraint of CTL that requires every $\mathbf{U}$ to be immediately quantified by $\forall$ or $\exists$, so that in CTL* one can write formulae such as $\forall[(\phi\,\mathbf{U}\,\psi)\vee\neg(\phi'\,\mathbf{U}\,\psi')]$. Consequently, not all CTL* formulae have meanings purely in terms of states, and the question of translation into a purely state-based logic like $L\mu$ becomes problematic. However, one can ask the question, is every state formula of CTL* (that is, boolean combinations of atoms and quantified formulae) equivalent to an $L\mu$ formula? The answer is 'yes', but it is a harder problem. Wolper, in an unpublished note from the early 1980s, noted that state formulas of CTL* can be translated via automata theory into PDL over a single label with looping (which, in turn, is directly translatable into $L\mu$). The first explicit translation was given by Dam [17], but the translation is very difficult, and gives a doubly exponential blowup in the formula size. The latter means that the translation is of no use for model-checking, as existing CTL* algorithms are much faster than a double exponential blowup of $L\mu$ model-checking. A few years later, Bhat and Cleaveland [6] gave a single exponential translation into the equational variant of $L\mu$. Although still quite complex, utilising a so-called Büchi tableau automaton as an intermediary, this translation is efficient enough to give competitive model-checking of CTL* via translation.

4.2 Bisimulation and tree model property

Bisimulation or back-and-forth equivalence or zig-zag equivalence is the equivalence associated with modal logic. In our setting, a *bisimulation* between two $L\mu$ structures $\mathfrak{T}_1$ and $\mathfrak{T}_2$ over the same proposition set Prop and label set $\mathcal{L}$ is a relation R such that for all propositions P, if $P(s_1)$ and $s_1 R s_2$, then $P(s_2)$, and conversely; and if $s_1 R s_2$, and $s_1 \xrightarrow{a} s_1'$, then for some s_2', $s_2 \xrightarrow{a} s_2'$ and $s_1' R s_2'$, and conversely. Two states s_1 and s_2 are *bisimilar* if there is some bisimulation R such that $s_1 R s_2$.

Recall that HML is the fixpoint-free part of $L\mu$. The following is easily shown by structural induction on formulae:

THEOREM 3. If two states (in the same or different systems) are bisimilar, they satisfy the same HML formulae.

By an induction on approximants, it is also straightforward to extend this to

THEOREM 4. If two states (in the same or different systems) are bisimilar, they satisfy the same $L\mu$ formulae.

A system is *image-finite* if for all states s and labels a, the set $\{\, s' \mid s \xrightarrow{a} s' \,\}$ is finite. The following theorem holds:

THEOREM 5. If two states in image-finite systems satisfy the same HML (or $L\mu$) formulae, then they are bisimilar.

To prove this, one observes that bisimulation itself is a maximal fixpoint, namely the maximal fixpoint of the map $R \mapsto \{\, (s_1, s_2) \mid \big(s_1 \xrightarrow{a} s_1' \Rightarrow \exists s_2'. s_2 \xrightarrow{a} s_2' \wedge (s_1', s_2') \in R\big) \wedge \big(s_2 \xrightarrow{a} s_2' \Rightarrow \exists s_1'. s_1 \xrightarrow{a} s_1' \wedge (s_1', s_2') \in R\big) \,\}$ (ignoring the propositions, which can be dealt with by an additional clause); shows that in an image-finite system the approximants to this fixpoint close at ω; and then deduce that if two states are not bisimilar, there is a finite modal formula distinguishing them. The latter theorem does not hold for general systems: there are systems which satisfy the same $L\mu$ formulae, but are not bisimilar. The following example is based on one by Roope Kaivola. Let

$\phi_1, \phi_2, \ldots$ be an enumeration of all $L\mu$ formulae over some finite label set $\mathcal{L}$. Let $\mathfrak{T}_i$, with initial state s_i, be a finite model for ϕ_i, with all $\mathfrak{T}_i$ disjoint. Let $\mathfrak{T}_0$ be constructed by taking an initial state s_0 and making $s_0 \xrightarrow{a} s_i$ for all $i > 0$. Let $\mathfrak{T}'_0$ be $\mathfrak{T}_0$ with the addition of a transition $s_0 \xrightarrow{a} s_0$. $\mathfrak{T}_0$ and $\mathfrak{T}'_0$ are clearly not bisimilar, because in $\mathfrak{T}'_0$ it is possible to defer indefinitely the choice of which $\mathfrak{T}_i$ to enter. On the other hand, suppose that ψ is a formula, and w.l.o.g. assume the topmost operator is a modality. If the modality is $[b]$, ψ is true of both $\mathfrak{T}_0$ and $\mathfrak{T}'_0$; if it is $\langle b \rangle$, ψ is false of both; if ψ is $\langle a \rangle \psi'$, then ψ is false at both $\mathfrak{T}_0$ and $\mathfrak{T}'_0$ iff ψ' is unsatisfiable, and true at both otherwise; if ψ is $[a]\psi'$, then ψ is true at both $\mathfrak{T}_0$ and $\mathfrak{T}'_0$ iff ψ' is valid, and false at both otherwise.

A simple corollary of theorem 4 is that $L\mu$ has the *tree model property.*

Proposition 6. If a formula has a model, it has a model that is a tree.

Just unravel the original model, thereby preserving bisimulation. This can be strengthened to the *bounded branching degree* tree model property (just cut off all the branches that are not actually required by some diamond subformula; this leaves at most (number of diamond subformulae) branches at each node).

For a more detailed look at bisimulation, see Chapter 5 of this Handbook.

4.3 $L\mu$ and automata

$L\mu$ is intimately related to automata theory, and the equivalence between various automata, particularly alternating parity automata, as described in section 5, and $L\mu$ is a key technique in many results. The first connexion between $L\mu$ and automata was tree automata, which we now briefly review.

Let us start with the notion of an automaton familiar from introductory computer science courses. A finite automaton $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ consists of a finite set of states Q, a finite alphabet Σ, a transition function δ, an initial state $q_0 \in Q$ and an acceptance condition F. Classical nondeterministic automata recognise languages, subsets of Σ^*, where the transition function $\delta : Q \times \Sigma \to \wp Q$. Given a word $w = a_0 \ldots a_n \in \Sigma^*$, a *run* of $\mathcal{A}$ on w is a sequence of states $q_0 \ldots q_n$ that traverses w, so $q_{i+1} \in \delta(q_i, a_{i+1})$. The run is *accepting* if the sequence $q_0 \ldots q_n$ obeys F: classically, $F \subseteq Q$ and $q_0 \ldots q_n$ is accepting if the last state $q_n \in F$. There may be many different runs of $\mathcal{A}$ on w, some accepting the others rejecting, or no runs at all. The language *recognised* by $\mathcal{A}$ is the set of words for which there is at least one accepting run. A simple extension is to allow recognition of infinite length words. A run of $\mathcal{A}$ on $w = a_1 \ldots a_i \ldots$ is an infinite sequence of states $\pi = q_0 \ldots q_i \ldots$ that travels over w, so $q_{i+1} \in \delta(q_i, a_{i+1})$ and it is accepting if it obeys the condition F. Let $\mathit{inf}(\pi) \subseteq Q$ contain exactly the states that occur infinitely often in π. Classically, $F \subseteq Q$ and π is accepting if $\mathit{inf}(\pi) \cap F \neq \emptyset$ which is the Büchi acceptance condition.

Büchi automata are an alternative notation for characterizing infinite runs of systems. Assume Prop is a finite set. The alphabet $\Sigma = \wp\mathrm{Prop}$. If $\pi = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \ldots$ is an infinite run, then $\pi \models \mathcal{A}$ if the automaton accepts the word $\mathrm{Prop}(s_0)\,\mathrm{Prop}(s_1) \ldots$ where $\mathrm{Prop}(s_i)$ is the subset of Prop that is true at s_i. For example, if $\mathrm{Prop} = \{P\}$, $Q = \{p, q\}$, $\delta(p, \{P\}) = \{q\}$, $\delta(p, \emptyset) = \{p\}$, $\delta(q, \{P\}) = \{q\}$ and $\delta(q, \emptyset) = \{q\}$, $q_0 = p$ and $F = \{q\}$, then this automaton is equivalent to the temporal formula $\mathbf{F}P$. (In fact, Büchi automata coincide with the *linear-time μ-calculus* where fixpoints are added to simple next time temporal logic that has the sole modality $\bigcirc$.)

When formulae are equivalent to automata, satisfiability checking reduces to the *non-emptiness* problem for the automata: that is, whether the automaton accepts something. If $\mathcal{A}$ is a Büchi automaton, then it is non-empty if there is a transition path $q_0 \longrightarrow^* q \in F$ and a cycle $q \longrightarrow^* q$ (equivalent to an eventually cyclic model).

The notion of bounded branching *tree automaton* extends the definition of automaton to accept n-branching infinite trees whose nodes are labelled with elements of Σ. Previously, states q' belonged to $\delta(q, a)$; now it is tuples $(q'_1, \ldots, q'_n)$ that belong to $\delta(q, a)$. A tree automaton traverses the tree, descending from a node to all n-child nodes, so the automaton splits itself into n copies, and proceeds independently. A run of the automaton is then an n-branching infinite tree labelled with states of the automaton. A run is accepting if *every* path through this tree satisfies the acceptance condition F. In the case of Rabin acceptance $F = \{(G_1, R_1), \ldots (G_k, R_k)\}$ where each $G_i, R_i \subseteq Q$ and π obeys F if there is a j such that $\mathit{inf}(\pi) \cap G_j \neq \emptyset$ and $\mathit{inf}(\pi) \cap R_j = \emptyset$. A variant definition is *parity* acceptance first introduced (not under that name) by Mostowski [52] where F maps each state q of the automaton to a *priority* $F(q) \in \mathbb{N}$. We say that a path satisfies F if the least priority seen infinitely often is even. It is not hard to see that a parity condition is a special case of a Rabin condition; it is also true, though somewhat trickier, that a Rabin automaton can be translated to an equivalent parity automaton.

Assuming Prop is finite, tree automata characterize rooted n-branching infinite tree models $(\mathfrak{T}, s_0, \mathfrak{V}_{\mathrm{Prop}})$ for $L\mu$ formulae where s_0 is the root of the tree: $(\mathfrak{T}, s_0, \mathfrak{V}_{\mathrm{Prop}}) \models \mathcal{A}$ if $\mathcal{A}$ accepts the behaviour tree $\mathfrak{T}'$ that replaces each state $s \in \mathfrak{T}$ with $\mathrm{Prop}(s)$. For example, let $\mathrm{Prop} = \{P\}$, $Q = \{p, q\}$, $\delta(p, \{P\}) = \{(p, p)\}$, $\delta(p, \emptyset) = \{(q, q)\}$, $\delta(q, \{P\}) = \{(p, p)\}$ and $\delta(q, \emptyset) = \{(q, q)\}$ and $q_0 = p$. This automaton $\mathcal{A}$ with parity acceptance $F(p) = 1$ and $F(q) = 2$ is equivalent to $\mu Y. \nu Z. (P \wedge [a]Y) \vee (\neg P \wedge [a]Z)$ over infinite binary-tree models: the fixed point μY is 'coded' by p and νZ is coded by q.

The use of priorities looks very much like the definition of well-founded pre-model from section 3.6, if we assign priorities to the subformulae of an $L\mu$ formula in such a way that the priority of a fixpoint formula is lower than any of its subformulae (and the priority of a least fixpoint is odd). Indeed, it is essentially the same thing. Tree automata and $L\mu$ are equivalent [64]:

THEOREM 7. A family of n-branching infinite tree models is defined by some tree automaton iff it is the set of n-branching infinite tree models of some corresponding $L\mu$ formula. Consequently, determining whether a system satisfies an $L\mu$ formula is equivalent to determining whether its behaviour trees are accepted by the corresponding automaton.

Decidability of satisfiability of $L\mu$ formulae reduces to the non-emptiness problem for tree automata. This problem is more difficult than for Büchi automata. However, there is an exponential decision procedure that is inductive in the *index* of the automaton (which is the number of parities or pairs in F, in the case of a Rabin automaton).

This illustrates the potency of the automata-theoretic approach to temporal logic that has become popular in recent years. Satisfiability of formulae is reduced to the non-emptiness problem for a class of automata. There is also the virtue that automata sustain combinatorial transformations, such as determinization, and closure operations, such as intersection, that are not in the logical repertoire. Occasionally, logics are easier: one of the hardest automata-theoretic proofs is that tree automata are closed under complementation. We shall see more of automata in later sections.

4.4 $L\mu$ and games

$L\mu$ is also intimately related to games, as are automata. We can view the relationship at different levels.

The fundamental semantic theorem can be presented as a simple two player *model-checking* game. Assume a rooted model $(\mathfrak{T}, s_0, \mathfrak{V})$ and formula ϕ_0 in positive normal form. The game $G(s_0, \phi_0)$ is defined on an *arena* that is a set of pairs (s, ψ) where s is a state of $\mathfrak{T}$ and ψ is a subformula of ϕ_0. The initial position is (s_0, ϕ_0). There are two players, whom we will call simply $\forall$ and $\exists$. (Other popular names include Player II/Player I, Abelard/Eloise, Opponent/Proponent, Refuter/Verifier.) $\forall$ is responsible for making a move from a position $(s, \phi \wedge \psi)$, the available choices are $\{(s, \phi), (s, \psi)\}$, and from a position $(s, [a]\phi)$ whose available choices are $\{(t, \phi) \mid s \xrightarrow{a} t \in \mathfrak{T}\}$. Similarly, $\exists$ is responsible for $(s, \phi \vee \psi)$ and $(s, \langle a \rangle \phi)$. There are final positions (s, ψ) where $\psi \in \{P, \neg P, [a]\phi, \langle a \rangle \phi\}$: $(s, [a]\psi)$ and $(s, \langle a \rangle \phi)$ are only final if there is no state t such that $s \xrightarrow{a} t$. A final position (s, ψ) is winning for $\exists$ if $s \models \psi$; otherwise it is winning for $\forall$.

A play of $G(s_0, \phi_0)$ is a finite or infinite sequence of positions starting with (s_0, ϕ_0). $\exists$ wins a finite play if the final position is winning for $\exists$. She wins an infinite play if the outermost fixed point variable Y that occurs infinitely often in the play is a ν-variable. Otherwise, $\forall$ wins. There may be many different plays; $\exists$ may win some and lose others. A *strategy* for a player is a function which, given a play so far and a position where there is a choice, returns a specific choice and so tells the player how to move. A *history-free* (positional or memoryless) strategy only depends on the current position and not on the previous history of the play: for $\exists$ it is just a choice function. A *winning strategy* is one which, if followed, guarantees that the player will win all plays of the game. Now the fundamental semantic theorems, theorems 1 and 2, are equivalent to the following.

THEOREM 8. $s \models \phi$ iff $\exists$ has a history-free winning strategy for the game $G(s, \phi)$.

The model checking game on finite structures can be abstracted into a simple two player graph game, called *the parity game*. The state set Q of the graph are the positions and are partitioned into $Q_\forall$ and $Q_\exists$. There is an initial state $q_0 \in Q$. Edges decide possible next positions; for instance, $\exists$ chooses a successor from a vertex $q \in Q_\exists$ and to ensure play is always infinite winning positions have self-loops. The acceptance condition F is just given as a parity condition: F maps each state q of the automaton to a priority $F(q) \in \mathbb{N}$ and $\exists$ wins an infinite play if the least priority that occurs infinitely often is even. The *model-checking* problem for $L\mu$ over finite structures, whether $s_0 \models \phi_0$, is equivalent to solving the parity game (does $\exists$ win q_0 ?). Parity games are determined (i.e. either $\exists$ or $\forall$ has a winning strategy), and a winning strategy is history-free. However, the exact complexity of solving a parity game is a significant open problem.

There is a more intimate connection between $L\mu$ and parity games. An $L\mu$ formula, itself, *is* a parity game as we shall see in section 5; alternating automata are games. Tree automata are games following Gurevich and Harrington [30]. Consider a run of a tree automaton on an n-branching infinite tree whose nodes are labelled with elements of Σ. It starts at the root of the tree with the initial automaton state. If the automaton is in state q examining a node v of the tree labelled with $a \in \Sigma$ then $\exists$ chooses a tuple $(q'_1, \ldots, q'_n)$ that belong to $\delta(q, a)$. Now $\forall$ chooses a direction $i : 1 \leq i \leq n$ so the next position is the ith child of v in state q'_i. The play continues forever. The play is won by $\exists$ if it obeys the acceptance condition. Clearly, $\exists$ has a winning strategy, iff the automaton accepts the tree. (If the acceptance condition is a Rabin condition, this strategy is not

history-free; however, it only depends on the 'latest appearance record', an ordering of the automaton states capturing the last time each automaton state occurred in the current play.)

5 DECIDABILITY OF SATISFIABILITY

As with any logic, a key question is decidability of satisfiability, that is, deciding whether a closed formula has a model. A connected property is the *finite model property (fmp)*, that is, if a formula has a model, then it has a finite model. If a logic has the fmp (and the size of the finite model for a formula is effectively bounded), then decidability follows, since one can just check all models up to the size bound. $L\mu$, as we have seen, has the tree model property.

A direct approach to proving decidability of satisfiability is to employ semantic tableaux, to begin with an initial closed formula ϕ in positive normal form and then to build a tree model for it whose states are labelled with locally consistent subsets of the Fisher-Ladner closure of ϕ, $\mathrm{cl}(\phi)$: for instance, if $\psi \wedge \gamma \in s$ then $\psi \in s$ and $\gamma \in s$. Children of a node s are generated using modal successor principles. For each $\langle a \rangle\psi \in s$ create a child node t such that $s \xrightarrow{a} t$ and $\psi \in t$: in turn, $s \xrightarrow{a} t$ when $[a]\psi \in s$ implies $\psi \in t$ and $\psi \in t$ and $\langle a \rangle\psi \in \mathrm{cl}(\phi)$ implies $\langle a \rangle\psi \in s$. This guarantees that the tree has bounded branching degree because $\mathrm{cl}(\phi)$ is finite. Fixed point formulae are "unfolded": ${}^{\mu}_{\nu}X.\,\psi \in s$ implies $\psi({}^{\mu}_{\nu}X.\,\psi) \in s$. The valuation $\mathfrak{V}_{\mathrm{Prop}}$ is then defined: $s \in \mathfrak{V}_{\mathrm{Prop}}(P)$ if and only if $P \in s$.

If ϕ is satisfiable then the construction will generate a finite tree model or an infinite tree that is a pre-model. In the latter case, the problem is how to ensure that it is well-founded. So far, there is no distinction between least and greatest fixed points. As mentioned, an important semantic principle is Park's fixed point induction rule, if $\models \phi(\psi) \Rightarrow \psi$ then $\models \mu X.\phi(X) \Rightarrow \psi$: it follows directly from the semantics because μ is indeed the least pre-fixed point. A question is how to use this semantic principle to guide the tableau construction in such a way that if the starting formula is satisfiable then a model is generable. The following proposition is useful.

Proposition 9. If $\gamma \wedge \mu X.\psi(X)$ is satisfiable and X is not free in γ, then $\gamma \wedge \psi(\mu X.\neg\gamma \wedge \psi)$ is satisfiable.

Proof. Assume that $\gamma \wedge \mu X.\psi(X)$ is satisfiable but $\models \psi(\mu X.\neg\gamma \wedge \psi) \Rightarrow \neg\gamma$. Therefore, $\models \psi(\mu X.\neg\gamma \wedge \psi) \Rightarrow \neg\gamma \wedge \psi(\mu X.\neg\gamma \wedge \psi)$. Using the fact that $\models \phi'(\mu X.\phi'(X)) \Rightarrow \mu X.\phi'(X)$ and propositional reasoning, $\models \psi(\mu X.\neg\gamma \wedge \psi) \Rightarrow \mu X.\neg\gamma \wedge \psi$. By fixed point induction, $\models \mu X.\psi \Rightarrow \mu X.\neg\gamma \wedge \psi$ and consequently $\models \mu X.\psi \Rightarrow \neg\gamma$ which is a contradiction. ❑

5.1 *The aconjunctive fragment*

The tableau approach was employed by Kozen [35] to decide satisfiability. Unfortunately, he could only prove the result for a sublogic of $L\mu$, when the starting formula ϕ is *aconjunctive*: that is, if $\mu X.\psi$ is a subformula of ϕ and $\psi_1 \wedge \psi_2 \in \mathrm{cl}(\mu X.\psi)$ then for at most one ψ_i is it the case that $\mu X.\psi \in \mathrm{cl}(\psi_i)$. For instance, $\nu Z.\mu X.([a]X \vee \langle b \rangle Z) \wedge \langle a \rangle Z$ is aconjunctive: the subformula $\gamma = \mu X.([a]X \vee \langle b \rangle Z) \wedge \langle a \rangle Z$ has one conjunction in its closure $([a]\gamma \vee \langle b \rangle Z) \wedge \langle a \rangle Z$ and γ is only in the closure of the first conjunct. In contrast, $\gamma = \mu X.\nu Z.([a]X \vee \langle b \rangle Z) \wedge \langle a \rangle Z$ fails to be aconjunctive: γ is in the closure of both conjuncts $([a]\gamma \vee \langle b \rangle(\nu Z.[a]\gamma \wedge \langle a \rangle Z)) \wedge \langle a \rangle(\nu Z.[a]\gamma \wedge \langle a \rangle Z)$. Aconjunctivity restricts

how a formula $\mu X.\psi \in \mathrm{cl}(\phi)$ can regenerate itself in the tableau construction: there can only be a linear pattern of regeneration (as opposed to the more general branching pattern for full $L\mu$). In the case of $\gamma = \nu Z.\mu X.([a]X \vee \langle b\rangle Z) \wedge \langle a\rangle Z$, the relevant formula $\gamma' = \mu X.([a]X \vee \langle b\rangle \gamma) \wedge \langle a\rangle \gamma$ can only regenerate itself through the the subformula $[a]X$: so, multiple regenerations of γ' happen only as part of a linear sequence. On the other hand, $\gamma = \mu X.\nu Z.([a]X \vee \langle b\rangle Z) \wedge \langle a\rangle Z$ can regenerate itself through both subformulae $([a]X \vee \langle b\rangle Z)$ and $\langle a\rangle Z$: so, multiple regenerations of γ form a tree.

Given the aconjunctive restriction, one can guide the construction of the tree model by applying proposition 9 to $\mu X.\psi \in s$: as it is unfolded its interpretation is strengthened to $\psi(\mu X.\neg s \wedge \psi)$ where s abbreviates the conjunction of all formulas in s. The strengthening interpretation is extended as $\mu X.\psi$ regenerates itself in descendent states t of s, so that an unfolding in t is re-interpreted as $\psi(\mu X.\neg s \wedge \ldots \wedge \neg t \wedge \psi)$ thereby ensuring that a descendent state within which $\mu X.\psi$ is regenerated cannot have the same labelling as the ascendent state (and because the starting formula is aconjunctive this will guarantee a well-founded pre-model). To do this, one needs a careful ordering on fixed point subformulae (in terms of which are more outermost) to ensure that the set of labellings remains finite. Kozen showed that the decision procedure for this fragment (that contains PDL and CTL) works in exponential time and at the same time the proof delivers the finite model property. In fact, the construction works for a more generous fragment of the logic, called the *weak* aconjunctive fragment in [71]. One only needs to guarantee that there is a linear pattern of regeneration of least fixed point subformulae relative to each individual branch in the tree model. The formula $\gamma = \mu X.\nu Z.\langle a\rangle X \wedge \langle a\rangle Z$ belongs to this more generous fragment because the regenerations of γ through the subformulae $\langle a\rangle X$ and $\langle a\rangle Z$ happen in different branches: the formula $\mu X.\nu Z.([a]X \vee \langle b\rangle Z) \wedge \langle a\rangle Z$ does not belong to it. In fact, every closed formula of $L\mu$ is semantically equivalent to a weak aconjunctive formula (which follows from results below). However, it is an open question whether the tableau technique can be made to work directly for all formulae.

5.2 Towards automata

The first decision procedure for full $L\mu$ reduced the problem to SnS, the second-order theory of n-successors, [38]. The structure for SnS is the transition system (tree) with state space $\{0, \ldots, n-1\}^*$ and transition relations $w \xrightarrow{i} wi$ for each $i < n$. Büchi showed that the monadic second-order (MSO) theory of S1S is decidable [13]: besides first-order constructs, MSO has quantifiers over sets of states. S1S is a weak form of arithmetic where, in this presentation, the number n is 0^n and $\xrightarrow{0}$ is the $+1$ function. Rabin extended Büchi's result by showing that the MSO theory of SnS is decidable for any $n > 0$ [60]. Kozen and Parikh's proof of decidability of satisfiability for full $L\mu$ uses the tree model property with bounded branching degree. Given a formula ϕ the maximum required branching degree n can be calculated from $\mathrm{cl}(\phi)$. The formula ϕ can then be translated almost directly into SnS by examining its semantics: for instance, $\|\nu X.\phi\|_{\mathfrak{V}} = \exists S.S \subseteq \|\phi\|_{\mathfrak{V}[X:=S]}$ and $\|X\|_{\mathfrak{V}} = \mathfrak{V}(X)$. Elements of Prop are treated as variables (and are existentially quantified over). Labels in diamond modalities are judiciously mapped to "directions" $i < n$ and labels in box modalities to sets of directions. For example, $\nu X.\langle a\rangle(X \wedge \neg P) \wedge \langle a\rangle(X \wedge P)$ is translatable into the S2S formula

$$\exists P.\exists S.\forall x.\exists y.\exists z.(x \in S \Rightarrow x \xrightarrow{1} y \wedge y \in S \wedge y \notin P) \wedge (x \in S \Rightarrow x \xrightarrow{2} z \wedge z \in S \wedge z \in P)$$

The formula ϕ is satisfiable if and only if its translation is true in SnS: for instance, the S2S formula above is true. The key feature in the MSO decidability proofs is that in a formula $\exists X.\phi$, quantification can be restricted to "regular" sets of states which leads to quantifier elimination when the normal form is a nondeterministic finite state automaton. In the case of S1S it is a Büchi word automaton and in the more general setting of SnS it is a Rabin tree automaton: these automata are defined in section 4.3. The automaton normal form for $\forall X.\phi$, that is $\neg\exists X\neg\phi$, involves an exponential increase in size because of complementation. Consequently, the decision procedure for SnS, $n > 0$, is (and must be) non-elementary. Because $\mu X_1.\nu X_2.\ldots\mu X_m.\nu X_{m+1}.\psi$ is translated into the MSO formula $\forall S_1.\exists S_2\ldots\forall S_m.\exists S_{m+1}.\psi'$, Kozen and Parikh's decision procedure for $L\mu$ is also non-elementary.

MSO formulae with second-order quantification, unlike fixed point formulae, are expressively succinct. A direct translation of $L\mu$ formulae into finite state automata, without intervening MSO formulae, could lead to a more efficient decision procedure. With this technique Streett provided an elementary time decision procedure for PDL with looping and converse [65]. With Emerson he employed the same technique for $L\mu$ and obtained a decision procedure for satisfiability and a proof of the finite model property at once [64]. The procedure is in elementary time. The central ingredient (besides the tree model property) is the relationship between $L\mu$ and Rabin automata, which is established using the fundamental semantic theorem. For, the constraint on fixpoint regeneration and infinite repetition is expressible as a Rabin acceptance condition. Now we can construct an automaton that accepts such bounded-branching tree models, by combining a finite-state automaton to check the local consistency (that is, to check that the putative model is a pre-model), and a Rabin automaton to check that the pre-model is well-founded. Thus the formula is satisfiable if this product automaton accepts some tree. Now automata theory, see for instance [66], tells us that (a) this question is decidable (b) if such an automaton accepts some tree, it accepts a regular tree, that is, one that is the unravelling of a finite system; this gives the results. Later, Emerson and Jutla provided an exponential time decision procedure (which is optimal) using an improved determinization construction and an improved tree automata emptiness test [22]: there is an exponential (in the size of the formula) bound on the size of the model.

5.3 Alternating parity automata

There is a slight mismatch between $L\mu$ models and SnS models because of the fixed branching degree and the explicit indexed successors. However, it is possible to define automata that can directly recognise $L\mu$ models by navigating through their transition graphs. We define *alternating parity automata* for this purpose following, for example, [40]. The only restriction is that we assume that Prop is a finite set (those propositions that appear in a starting formula ϕ). A rooted model for a closed formula ϕ is a triple $(\mathfrak{T}, s_0, \mathfrak{V})$.

Recall the notion of automaton $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ as defined in section 4.3. Now think of the transition function of an automaton as a logical formula. For a word automaton, if $\delta(q, a) = \{q_1, \ldots, q_m\}$ then it is the formula $q_1 \vee \ldots \vee q_n$. For a tree automaton if $\delta(q, a) = \{(q_1^1, \ldots, q_n^1), \ldots, (q_1^m, \ldots, q_n^m)\}$ then it is $((1, q_1^1) \wedge \ldots \wedge (n, q_n^1)) \vee \ldots \vee ((1, q_1^m) \wedge \ldots \wedge (n, q_n^m))$: here the element (i, q') means create an *ith*-child with label q'. A word or tree is accepted if there exists an accepting run for that word or tree; hence, the disjuncts.

However, for a tree, every path through it must be accepting; hence the conjuncts. In *alternating* word automata, the transition function is given as an arbitrary boolean expression over states: for instance, $\delta(q,a) = q_1 \wedge (q_2 \vee q_3)$. In alternating tree automata it is a boolean expression over directions and states: for instance, $((1,q_1) \wedge (1,q_2)) \vee (2,q_3)$. Now the definition of a run becomes a tree in which, successor transitions obey the boolean formula. In particular, even for an alternating automaton on words, a run is a tree, and not just a word. The acceptance criterion is as before, that every path of the run must be accepting. An alternating automaton is just a game too where $\forall$ is responsible for $\wedge$ choices and $\exists$ for $\vee$ choices (as in section 4.4).

The idea now is to replace pairs (i,q') with simple modal formulae. We define *modal* automata whose transition functions appeal to a modal language (similar to modal equation systems). Formally, a modal automaton $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ is the set $\wp$Prop and F is the parity acceptance condition. The transition function $\delta : Q \times \Sigma \rightarrow B$ where B is the following set of positive modal formulae (with modal depth at most 1).

- $\top$, $\perp$ are in B
- If $q \in Q$ and $a \in \mathcal{L}$ then $\langle a \rangle q$ and $[a]q$ are in B
- If B_1 and B_2 are in B then $B_1 \vee B_2$ and $B_1 \wedge B_2$ are in B

The automaton traverses the modal model, starting at s_0 and moving from a state s to successor states t such that $s \xrightarrow{a} t$ for some $a \in \mathcal{L}$, according to the transition function. However, not every successor may be included and some successors may be included multiple times: for instance, if q is the current automaton state for s and $\delta(q, \mathrm{Prop}(s)) = \langle a \rangle p_1 \wedge [c]p_2$[3] and $s \xrightarrow{a} s_1$, $s \xrightarrow{b} s_2$, $s \xrightarrow{c} s_1$ and $s \xrightarrow{c} s$ then the automaton changes state to p_1 and moves to s_1 in the model, changes to p_2 and also moves to s_1 in the model and changes to p_2 and remains at s. As with tree automata, a run of $\mathcal{A}$ on a model is a a labelled tree $(N, \xrightarrow{i}, L')$ where $N \subseteq \omega^*$ that obeys the tree property that if $wi \in N$ then $w \in N$ and $w \xrightarrow{i} wi$: a node $x \in N$ may have infinitely many successors $xi \in N$, as models have no bound on their branching degree. Unlike tree automata, there is no requirement that complete branches should have infinite length.

In more detail, a run of $\mathcal{A}$ on a modal model is a projection of an intermediate structure, a tree with composite labels $(N, \xrightarrow{i}, L)$. The labelling $L : N \rightarrow S \times Q$ where S is the state space of the model: for the root of the tree, $L(\epsilon) = (s_0, q_0)$. The labels of a node and its successors have to obey the transition function. First, given a state s of the model let M_s range over mixture subsets $\{(t,q') \mid s \xrightarrow{a} t \text{ for some } a \text{ and } q' \in Q\}$. Next, we define when a subset M_s satisfies a modal formula B, which we write $M_s \models B$.

$$
\begin{array}{lll}
M_s \models \top & & M_s \not\models \perp \\
M_s \models \langle a \rangle p & \text{iff} & \exists t. s \xrightarrow{a} t \text{ and } (t,p) \in M_s \\
M_s \models [a]p & \text{iff} & \forall t. \text{ if } s \xrightarrow{a} t \text{ then } (t,p) \in M_s \\
M_s \models B_1 \vee B_2 & \text{iff} & M_s \models B_1 \text{ or } M_s \models B_2 \\
M_s \models B_1 \wedge B_2 & \text{iff} & M_s \models B_1 \text{ and } M_s \models B_2
\end{array}
$$

Given $\mathcal{A}$ and a rooted model, one grows a labelled tree from the root ϵ with $L(\epsilon) = (s_0, q_0)$. If $L(x) = (s,q)$ and $\delta(q, \mathrm{Prop}(s)) = B$ then there is a (possibly empty) set M_s such that

[3]Prop(s) is the subset of Prop true at s.

$M_s \models B$. A child of x is produced for each element of M_s: that is, $M_s = \{L(xi) \mid xi \in N\}$. For example, if $L(x) = (s, q)$ and $\delta(q, \mathrm{Prop}(s)) = \langle a\rangle q_1 \wedge [a]q_2 \wedge \langle a\rangle q_3$ and in the model $s \xrightarrow{a} s_i$ for all $i \geq 0$ then a candidate M_s is $\{(s_0, q_1), (s_1, q_3), (s_0, q_2), \ldots, (s_i, q_2), \ldots\}$: here there are infinitely many such candidates. The required run is the projection of the tree to states in Q, the tree $(N, \xrightarrow{i}, L')$ whose labelling $L'(x) = q$ if $L(x) = (s, q)$ for some s. A run is *accepting* if all (labellings of) infinite branches starting from the root obey the parity acceptance condition F.

Given a rooted model $(\mathfrak{T}, s_0, \mathfrak{V})$, $s_0 \models \mathcal{A}$ if there is an accepting run of $\mathcal{A}$ on that model. The following is relatively straightforward (and is reminiscent of translating to and from boolean equation systems).

THEOREM 10. For each modal automaton there is an equivalent closed formula of $L\mu$, and for each closed formula of $L\mu$ there is an equivalent modal automaton.

5.4 Automaton normal form

We can now extract a semantic *normal form* for $L\mu$ due to Walukiewicz [71, 32]. If Γ is a finite set of formulae, $(a)\Gamma$ abbreviates $\bigwedge_{\phi\in\Gamma}\langle a\rangle\phi \wedge [a]\bigvee_{\phi\in\Gamma}\phi$. Every modal automaton is equivalent to a *restricted* modal automaton. Let $\Sigma = \{a_1, \ldots, a_n\}$. The transition function is restricted: formulae of B are disjunctions of conjuncts of the form $(a_1)B_1 \wedge \ldots \wedge (a_n)B_n$ where each $B_i \subseteq Q \cup \{\top\}$. The proof of the following is far from trivial and depends on the combinatorial features of automata, especially determinization.

THEOREM 11. For each modal automaton there is an equivalent restricted modal automaton.

A formula is in *automaton normal form (anf)*[4], if it belongs to the following sublogic, where

- P, $\neg P$ and Z are anfs
- If ϕ_1 and ϕ_2 are anfs, so is $\phi_1 \vee \phi_2$
- If ϕ is an anf, then so are $\nu Z.\phi$, $\mu Z.\phi$
- If each Γ_i is a finite set of anfs and $a_i \neq a_j$ when $i \neq j$ and α^+ is a finite set of atomic propositions and their negations, then $(a_1)\Gamma_1 \wedge \ldots \wedge (a_n)\Gamma_n \wedge \alpha^+$ is an anf

Anf formulae are the characteristic formulae for restricted automata. For instance, a clause $\{\langle a\rangle p, \langle a\rangle q\}$ with respect to labels a, b becomes the formula $\langle a\rangle p \wedge \langle a\rangle q \wedge [a](p \vee q) \wedge [b]\bot$.

Proposition 12. For each restricted automaton there is an equivalent anf formula.

Therefore, anfs are semantic normal forms for $L\mu$. We can effectively construct the anf normal form for a formula ϕ in positive normal form. First, use Theorem 10 to build an equivalent modal automaton $\mathcal{A}_\phi$ for ϕ. Next, use Theorem 11 to transform $\mathcal{A}_\phi$ into an equivalent restricted automaton $\mathcal{A}^+_\phi$. Finally, use Proposition 12 to convert $\mathcal{A}^+_\phi$ into an equivalent anf formula ϕ'.

An anf formula is weakly aconjunctive (although not necessarily aconjunctive). After simplification, the anf normal form of the earlier formula $\mu X.\nu Z.([a]X \vee \langle b\rangle Z) \wedge \langle a\rangle Z$

[4] Walukiewicz terms them "disjunctive formulae".

that is not aconjunctive is $\mu X.\nu Z.(a)\{X\} \vee ((a)\{Z, \top\} \wedge (b)\{Z, \top\})$. In fact, conjunction is even more constraining in anf formulae. Consider, the semantic tableau construction for an anf formula ϕ. The only time we need to apply $\wedge$ decomposition is just before the application of modal successors: assume a state s is labelled with the formula $(a_1)\Gamma_1 \wedge \ldots \wedge (a_n)\Gamma_n \wedge \alpha^+ \in \mathrm{cl}(\phi)$. At s it reduces to its components $(a_1)\Gamma_1, \ldots, (a_n)\Gamma_n, \alpha^+$. If α^+ is consistent then modal children $s \xrightarrow{a_i} t$ are created: however, by the definition of (a) each modal successor t is labelled with a single anf formula in $\mathrm{cl}(\phi)$. Therefore, as shown by Walukiewicz, ϕ is satisfiable iff all its fixed point subformulae $\mu X.\psi(X)$ are replaced with $\psi(\bot)$ and all subformulae $\nu X.\psi(X)$ are replaced with $\psi(\top)$. To illustrate this, assume $\phi = \mu X.\psi$ is satisfiable. Consider a rooted model and a least ordinal o such that $s_0 \models \mu X^o.\psi$. Consider its semantic tableau with initial state s_0 labelled with ϕ. If there is a descendent state t that is also labelled with ϕ then $t \models \mu X^{o'}.\psi$ with $o' < o$ which contradicts that o is least. Therefore, there is a model for ϕ such that no descendent state is labelled with ϕ, which is, therefore, also a model for $\psi(\bot)$. Consequently, satisfiability checking for an anf formula can be done in linear time [32]. To obtain the fmp for anf ϕ, replace each subformulae $\mu X.\psi(X)$ with $\psi(\bot)$ and build a semantic tableaux for it. For modal successors, if at state s, $\nu X.\psi \in \Gamma$ and $(a)\Gamma \in s$ and some state t is on the path from the root to s and t is labelled with $\nu X.\psi$ then let $s \xrightarrow{a} t$: in this way, a finite model for ϕ is constructed.

6 COMPLETE AXIOMATIZATION

A related problem to decidability is the question of providing an axiomatization of the theory of the modal mu-calculus. In his original paper, Kozen presented the axiomatization as an equational theory which is equivalent to the following.

1. axioms and rules for minimal multi-modal logic K
2. $\phi(\mu X.\phi(X)) \Rightarrow \mu X.\phi(X)$
3. $\dfrac{\phi(\psi) \Rightarrow \psi}{\mu X.\phi(X) \Rightarrow \psi}$

Axiom 2 is the axiom for a least fixed point that its "unfolding" implies it and rule 3 is Park's fixed point induction rule. The duals of this axiom and rule for greatest fixed points are; $\nu X.\phi(X) \Rightarrow \phi(\mu X.\phi(X))$ and if $\psi \Rightarrow \phi(\psi)$ then $\psi \Rightarrow \nu X.\phi(X)$.

However, despite the naturalness of this axiomatization, Kozen was unable to show that it was complete. He was, however, able to show completeness for the aconjunctive fragment. In fact, a proof works for *weak* aconjunctive formulae using the consistency version of proposition 9: if $\gamma \wedge \mu X.\psi(X)$ is consistent[5] and X is not free in γ, then $\gamma \wedge \psi(\mu X.\neg\gamma \wedge \psi)$ is consistent. The proof is similar to the tableau construction described in section 5.1. Given an initial consistent formula in positive normal form one builds a tree model: the construction is guided by the proposition above as in the satisfiability proof.

[5]A formula ϕ is consistent with respect to an axiom system if $\phi \Rightarrow \bot$ is not derivable within the axiom system. Completeness of an axiom system is equivalent to the statement that every consistent formula has a model (is satisfiable).

Completeness for the full language remained open for more than a decade, until it was finally solved by Walukiewicz in [71], who established that Kozen's axiomatization is indeed complete. The proof is very involved and, in effect, internalises the automata theoretic satisfiability proof described earlier. It utilises automata normal form and weak aconjunctivity. It is more straightforward (as with satisifiability) to show using tableaux that if an anf formula is consistent then it has a model. Much harder to prove is that every (closed) formula is provably equivalent within the axiom system to an anf formula. Walukiewicz utilises games on infinite tableaux to show this.

The following are valid fixpoint principles (which, by duality also are true for ν).

$$\mu X.\mu Y.\phi(X,Y) \iff \mu X.\phi(X,X) \iff \mu Y.\mu X.\phi(X,Y)$$

Arnold and Niwinski call these "the golden lemma" of μ-calculus [5]. Other interesting valid fixpoint principles include $\mu X.\phi(X) \Rightarrow \nu X.\phi(X)$, by monotonicity, and the following, due to Niwinski, that generalises that 'almost always' implies 'infinitely often'.

$$\mu X.\nu Y.\phi(X,Y) \Rightarrow \nu Y.\mu X.\phi(X,Y)$$

Deriving these principles deductively from Kozen's complete axiom system is by no means easy (as opposed to their derivations using the semantics).

7 ALTERNATION

As we said earlier, the alternation of fixpoints is what gives $L\mu$ its expressive power, and also what appears to generate the computation complexity of model-checking. In this section, we study alternation in more detail. As we have said, the idea is to count alternations of minimal and maximal fixpoint operators, but to do so in a way that only counts real dependency. The paradigm is 'always eventually' versus 'infinitely often': the 'always eventually' formula

$$\nu Y.(\mu Z.P \vee \langle a\rangle Z) \wedge \langle a\rangle Y$$

is, using a straightforward model-checking algorithm, really no worse to compute than two disjoint fixpoints, since the inner fixpoint can be computed once and for all, rather than separately on each outer approximant; on the other hand, the 'infinitely often' formula

$$\nu Y.\mu Z.(P \vee \langle a\rangle Z) \wedge \langle a\rangle Y$$

really does need the full double induction on approximants.

The definition of Emerson and Lei takes care of this by observing that the 'eventually' subformula is a closed subformula, and giving a definition that ignores closed subformulae when counting alternations. The stronger notion of Niwiński, which also has the advantage of being robust under translation to modal equation systems, also observes that, for example, $\mu X.\nu Y.[-]Y \wedge \mu Z.[-](X \vee Z)$ although it looks like a $\mu/\nu/\mu$ formula, is morally a μ/ν formula, since the inner fixpoint does not refer to the middle fixpoint.

The alternation depth referred to in the complexity of model-checking is a measure of alternation that is symmetric in μ and ν. It is possible to give algorithms that compute the alternation depth of a formula [24, 1, 34], and this is how the notion was presented by

Emerson and Lei. However, for our purposes it is easier to start from a definition of classes for formula, formalizing the idea of 'a $\mu/\nu/\mu$ formula' etc.; such a definition is analogous to the usual definition of quantifer alternation for predicate logic, an analogy which will be exploited later. This was how Niwiński [53] presented the notion of alternation, and we follow his presentation.

Assuming positive form, a formula ϕ is said to be in the classes $\Sigma_0^{N\mu}$ and $\Pi_0^{N\mu}$ iff it contains no fixpoint operators. To form the class $\Sigma_{n+1}^{N\mu}$ (resp. $\Pi_{n+1}^{N\mu}$), take $\Sigma_n^{N\mu} \cup \Pi_n^{N\mu}$, and close under the following rules:

1. if $\phi_1, \phi_2 \in \Sigma_{n+1}^{N\mu}$ (resp. $\Pi_{n+1}^{N\mu}$), then $\phi_1 \vee \phi_2, \phi_1 \wedge \phi_2, \langle a\rangle\phi_1, [a]\phi_1 \in \Sigma_{n+1}^{N\mu}$ (resp. $\Pi_{n+1}^{N\mu}$);
2. if $\phi \in \Sigma_{n+1}^{N\mu}$ (resp. $\Pi_{n+1}^{N\mu}$), then $\mu Z.\phi \in \Sigma_{n+1}^{N\mu}$ (resp. $\nu Z.\phi \in \Pi_{n+1}^{N\mu}$);
3. if $\phi(Z), \psi \in \Sigma_{n+1}^{N\mu}$ (resp. $\Pi_{n+1}^{N\mu}$), then $\phi(\psi) \in \Sigma_{n+1}^{N\mu}$ (resp. $\Pi_{n+1}^{N\mu}$), *provided* that no free variable of ψ is captured by a fixpoint operator in ϕ.

If we omit the last clause, we get the definition of 'simple-minded' alternation $\Sigma_n^{S\mu}$, that just counts syntactic alternation; if we modify the last clause to read '... *provided* that ψ is a closed formula', we obtain the Emerson–Lei notion $\Sigma_n^{EL\mu}$. (We write just Σ_n^{μ} when the distinctions are not important, or when we are making a statement that applies to all versions.)

To get the symmetrical notion of *alternation depth* of ϕ, we can define it to be the least n such that $\phi \in \Sigma_{n+1}^{\mu} \cap \Pi_{n+1}^{\mu}$. To make these definitions clear, consider the following examples:

- The 'always eventually' formula is $\Pi_2^{S\mu}$, but not $\Sigma_2^{S\mu}$, and so its simple alternation depth is 2. However, in the Emerson–Lei notion, it is also $\Sigma_2^{EL\mu}$, since $\nu Y.W \wedge \langle a\rangle Y$ is $\Pi_1^{EL\mu}$ and so $\Sigma_2^{EL\mu}$, and by substituting the closed $\Sigma_2^{EL\mu}$ (and in fact $\Sigma_1^{EL\mu}$) formula $\mu Z.P \vee \langle a\rangle Z$ for W we get 'always eventually' in $\Sigma_2^{EL\mu}$; hence its Emerson–Lei (and Niwiński) alternation depth is 1.
- The 'infinitely often' formula is Σ_2^{μ} but not Π_2^{μ}, in all three definitions, and so has alternation depth 2.
- The formula $\mu X.\nu Y.[-]Y \wedge \mu Z.[-](X \vee Z)$ is $\Sigma_3^{S\mu}$, but not $\Pi_3^{S\mu}$; it is also $\Sigma_3^{EL\mu}$ but not $\Pi_3^{EL\mu}$, since there are no closed subformulae to bring the substitution clause into play. However, in the Niwiński definition, it is actually $\Sigma_2^{N\mu}$: $\nu Y.[-]Y \wedge W$ is $\Pi_1^{N\mu}$ and so $\Sigma_2^{N\mu}$; we can substitute the $\Sigma_1^{N\mu}$ formula $\mu Z.[-](X \vee Z)$ for W without variable capture, and so $\nu Y.[-]Y \wedge \mu Z.[-](X \vee Z)$ is $\Sigma_2^{N\mu}$; and now we can add the outer fixpoint, still remaining in $\Sigma_2^{N\mu}$.

A natural question is whether the hierarchy of properties definable by Σ_n^{μ} formulae is actually a strict hierarchy, or whether the hierarchy collapses at some point so that no further alternation is needed. This problem remained open for a while; by 1990, it was known that $\Sigma_2^{N\mu} \neq \Pi_2^{N\mu}$ [4]. No further advance was made until 1996, when the strictness of the hierarchy was established by Bradfield [8, 9, 11].

THEOREM 13. For every n, there is a formula $\phi \in \Sigma_n^{\mu}$ which is not equivalent to any Π_n^{μ} formula.

Bradfield established this for $\Sigma_n^{N\mu}$, which implies the result for the other two notions. At the same time, Lenzi [43] independently established a slightly weaker hierarchy theorem for $\Sigma_n^{EL\mu}$.

Lenzi's proof is technically complex, and the underlying stratagem is not easy. Bradfield's proof appears technically complex, but most of the complexity is really just routine recursion-theoretic coding; the underlying stratagem is quite simple, and in some ways surprising. If one takes first-order arithmetic, one can add fixpoint operators to it, and one can then define a fixpoint alternation hierarchy in arithmetic. A standard coding and diagonalization argument shows that this hierarchy is strict [9]. The trick now is to transfer this hierarchy to $L\mu$. Simply by writing down the semantics, it is clear (give or take some work to deal with the more complex notions of alternation) that if one takes a recursively presented transition system and codes it into the integers, then for a modal formula $\phi \in \Sigma_n^\mu$, its denotation $\|\phi\|$ is describable by an arithmetic Σ_n^μ formula. However, it is also possible, given any arithmetic fixpoint formula χ, to build a transition system and a modal formula ϕ, of the same alternation depth as χ, such that $\|\phi\|$ is characterized by χ. If we take χ to be a strict Σ_n^μ arithmetic formula, then no Π_n^μ arithmetic formula is equivalent to it, and therefore no Π_n^μ modal formula can be equivalent to ϕ. The transition system that is constructed is infinite, but by the finite model property, the hierarchy transfers down to the class of finite models.

Both proof techniques construct explicit examples of hard formulae. Bradfield's examples have the form

$$\mu X_n.\nu X_{n-1}.\ldots.\mu X_1.[c]X_1 \vee \langle a_1\rangle X_1 \vee \ldots \vee \langle a_n\rangle X_n.$$

There are further questions one can ask about the alternation hierarchy. For example, is it still strict even on the binary tree? The answer is yes, given independently by Bradfield [10, 11] and Arnold [3] – the latter also gives a nice alternative proof of the main theorem, using topological methods rather than computability methods.

A more interesting question, and one that is still open, is: given a formula, what is its 'semantic' alternation depth? That is, what is the least alternation depth of any equivalent formula? Otto [55] showed that it is decidable whether a formula is equivalent to an alternation-free formula, and then Küsters and Wilke showed [41] it for alternation depth 1. Decidability is not known for higher levels.

8 BISIMULATION INVARIANCE

A hallmark of modal logic is bisimulation invariance: if $s \models \phi$ and s and s' are bisimulation equivalent then $s' \models \phi$. As we have seen, this remains true for $L\mu$ formulae. In logic, in general, structures are viewed as equivalent when they are isomorphic. However, in computation when structures represent behaviour of systems weaker forms of equivalence, such as automata acceptance equivalence or bisimulation equivalence, are more appropriate; see, for example, Milner [51].

8.1 $L\mu$ and MSOL

A modal formula can be translated into an equivalent bisimulation invariant first-order logic formula (over transition graphs) with one free variable. The translation is merely

the semantics. Let $\phi[x]$ be the translation of ϕ with free variable x: for instance, $P[x] = P(x)$ and $\langle a \rangle \phi[x] = \exists y.\, x \xrightarrow{a} y \wedge \phi[y]$. Clearly, $s \models \phi$ iff $\phi[s]$ holds. Van Benthem proved the converse: a bisimulation invariant first-order logic formula with one free variable is equivalent to a modal formula. Modal logic is the bisimulation invariant fragment of first-order logic.

The question is whether there is a similar result for closed formulae of $L\mu$. As we have seen, there is an intimate relationship between $L\mu$ and automata, games or SnS. None of these notations provide an obvious semantics for $L\mu$ formulae. Monadic second-order logic (MSOL) of transition graphs extends first-order logic with quantification over monadic predicates. With this addition we can translate $L\mu$.

$$\nu Z.\phi[x] \quad = \quad \exists Z.(\forall y.Z(y) \Rightarrow \phi[y]) \Rightarrow Z(x)$$

So, an $L\mu$ formula is translated into an equivalent bisimulation invariant MSOL formula with one free variable. Remarkably, the converse is also true, as proved by Janin and Walukiewicz [33].

THEOREM 14. A bisimulation invariant MSOL formula with one free variable is equivalent to an $L\mu$ formula.

In other words, $L\mu$ is the bisimulation invariant fragment of MSOL.

The proof of this theorem is intricate and again illustrates the potency of automata. The authors define an *ω-expansion* of a rooted model which is like the usual unravelling of the system into a tree, with the addition that the tree contains ω-many copies of every successor node. If $\phi(x)$ is a bisimulation invariant MSOL formula and $\phi(s)$ holds where s is the root of a model then $\phi(s)$ remains true for the ω-expanded model.

The proof uses modal automata from section 5.3. The transition function is defined using a simple modal language. If the automaton is in state q and at state s in the modal model and $\delta(q, \mathrm{Prop}(s)) = B$ then there is a mixture set $M_s \models B$ where $M_s \subseteq \{(t, q') \mid s \xrightarrow{a} t$ for some a and $q' \in Q\}$. Instead of simple modal formulae B, the automaton could employ first-order formulae with one free variable $B[x]$. Now, for instance, $M_s \models \exists y.x \xrightarrow{a} y \wedge p[y]$ iff $(t, p) \in M_s$ for some t such that $s \xrightarrow{a} t$. Critically, there is also a similar automata characterisation of MSOL formulae on trees. The transition function $\delta : Q \times \Sigma \to B'$ where B' is very similar to $B[x]$ except that it involves inequalites. When in CNF, formulae $B'[x]$ have the form

$$\exists y_1, \ldots y_n.(\bigwedge_{i \neq j} y_i \neq y_j \wedge x \xrightarrow{a_1} y_1 \wedge p_1[y_1] \wedge \ldots \wedge x \xrightarrow{a_n} y_n \wedge p_n[y_n] \wedge \forall z. \bigwedge z \neq y_i \wedge \psi(z, x)$$

where $\psi(z, x)$ captures the "box" formulae. The inequalities are effectively redundant in an ω-expanded model. The formulae $B'[x]$ collapse to $B[x]$ with respect to these models.

Van Benthem's theorem also holds for finite models: modal logic is the bisimulation invariant fragment of first-order logic with respect to finite models. It is an open question if this is true for $L\mu$ and MSOL.

8.2 Multi-dimensional $L\mu$ and Ptime

A major interest is classifying logics according to their expressive power. Computationally, we can ask whether there are logics that characterize complexity classes. A classic

result is that existential second-order logic exactly captures NP properties of finite structures. A key open problem is whether there is such a logic for PTIME properties. (For finite structures *with a linear ordering* the PTIME properties are exactly captured by least fixed point logic of section 9.2.) However, Otto shows that *bisimulation invariant* monadic PTIME properties (of modal structures) is logically characterizable by a multi-dimensional $L\mu$ [54].

For simplicity, assume finite $L\mu$ rooted structures whose label set is a singleton and let Prop be finite. Formulas of $L\mu$ are interpreted with respect to a single state. Consider instead k-tuples of states $(s_1, \ldots, s_k)$. Given such tuples we can define transition relations $\xrightarrow{i}$, for each $i : 1 \leq i \leq k$: $(s_1, \ldots, s_k) \xrightarrow{i} (t_1, \ldots, t_k)$ if $s_i \longrightarrow t_i$ and $s_j = t_j$ for all $j \neq i$. Otto defines the logic $L\mu^k$ (with $L\mu = L\mu^1$). Formulae may contain variables x_i, $1 \leq i \leq k$. Atomic formulae have the form Px_i: $(s_1, \ldots, s_k) \models Px_i$ iff $P \in \mathrm{Prop}(s_i)$. Modal formulae have the form $\langle i \rangle \phi$ and $[i]\phi$. Formulae are closed under boolean connectives. There is a substitution operation $\sigma : \{1, \ldots, k\} \to \{1, \ldots, k\}$: $\phi\sigma$ is the formula $\phi\{x_{\sigma(1)}/x_1, \ldots, x_{\sigma(k)}/x_k\}$. Finally, fixed points are k-ary: $\mu X(x_1, \ldots, x_k).\phi$ (and are interpreted as in section 9.2). Formulae of $L\mu^k$ are bisimulation invariant. The logic that characterizes bisimulation invariant monadic PTIME are the monadic formulae of $\bigcup_{k>0} L\mu^k$. Crucially, for $k > 1$, bisimulation equivalence is definable in $L\mu^k$.

$$\nu X(x_1, \ldots, x_k). \bigwedge_{P \in \mathrm{Prop}} Px_1 \Leftrightarrow Px_2 \wedge [1]\langle 2 \rangle X(x_1, \ldots, x_k) \wedge [2]\langle 1 \rangle X(x_1, \ldots, x_k)$$

For canonical finite rooted models (rooted models quotiented with respect to bisimulation equivalence) one can define a linear ordering on states via bisimulation *in*equivalence. So, each PTIME property is definable in least fixed point logic and, in fact, in some $L\mu^k$.

8.3 Bisimulation quantifiers and interpolation

In previous sections, a number of standard logical questions about the $L\mu$ have been covered, such as satisfiability, completeness, etc. These were all addressed, if not solved, early in the history of the logic. There are other standard questions about logics which, perhaps surprisingly, were not addressed until quite recently. In this subsection, we describe briefly work on interpolation theorems and related issues. A key ingredient in these proofs is again alternating parity automata; another ingredient is an interesting notion of 'bisimulation quantifier'.

A logic enjoys the *Craig interpolation property* if whenever $\phi \Rightarrow \psi$, then there is a third formula χ, using only those atomic symbols occurring in both ϕ and ψ, such that $\phi \Rightarrow \chi \Rightarrow \psi$. The *uniform interpolation property* requires further that to find χ, it suffices to know only one of ϕ or ψ and what the common language is. (That is, one can construct the strongest formula implied by ϕ in a given language, or the weakest formula implying ψ in a given language.) Maksimova showed [47] that most common temporal logics do not have interpolation. In [16], d'Agostino and Hollenberg show that $L\mu$ has interpolation, and even uniform interpolation, as we now sketch.

Let ϕ be a sentence, and P an atomic proposition occurring in ϕ. The aim is to construct a formula $\widetilde{\exists} P.\phi$ which is the strongest implicate of ϕ in the language omitting P. This can be done by using results of the Janin–Walukiewicz paper discussed earlier: translate ϕ into an MSOL sentence $\tilde{\phi}$, quantify it (in MSOL) to form $\exists P.\tilde{\phi}$, and then apply the construction mentioned to produce again an $L\mu$ formula $(\exists P.\tilde{\phi})^\vee$, which is true in

any rooted structure whose ω-expansion satisfies $\exists P.\tilde{\phi}$; but if a structure satisfies ϕ, then its ω-expansion satisfies $\exists P.\tilde{\phi}$, since the original valuation of P provides a witness. With some more technical lemmas, it is shown that $(\exists P.\tilde{\phi})^{\vee}$ is indeed the uniform interpolant of ϕ with respect to the vocabulary omitting P, and this is the definition of $\widetilde{\exists} P.\phi$. A similar definition and construction also works for action labels: $\widetilde{\exists} a.\phi$ is the strongest implicate of ϕ in the language omitting the label a.

The reason for the notation $\widetilde{\exists} P.\phi$ is that from the construction, it can be seen that a rooted structure satisfies $\widetilde{\exists} P.\phi$ iff there is a bisimlation equivalent rooted structure in the vocabulary excluding P that satisfies ϕ.

In the above, $\widetilde{\exists} P.\phi$ was, by definition, an $L\mu$ formula. It is natural to ask whether bisimulation quantifiers can give the same expressive power as the fixpoint operators. It turns out to be not sufficient to add $\widetilde{\exists}$ to modal logic; but [16] does show that adding $\widetilde{\exists}$ to PDL gives $L\mu$.

The techniques used here also give further results. One of the most satisfying is a Lyndon theorem: if an $L\mu$ sentence is monotone in a proposition P, then it is equivalent to a sentence positive in P. The proof is intricate.

9 GENERALIZED MU-CALCULI

We have seen that $L\mu$ has many nice properties. One interesting thread of research in recent years has been the investigation of why it enjoys these properties – is it because it is a *modal* fixpoint logic, because it is a *fixpoint* logic, or what else? In this section, we will briefly survey some of these investigations, and some of the more interesting generalizations of $L\mu$.

9.1 $L\mu$ with past

A simple extension of $L\mu$ is to include converse labels $\overline{a}$: $t \xrightarrow{\overline{a}} s$ iff $s \xrightarrow{a} t$. Modalities can now include converses. $L\mu$ with converse fails to have the finite model property: $\nu X.\langle a\rangle(X \wedge \mu Y.[\overline{a}]Y)$ is only satisfiable in an infinite state model. However, it retains both the tree model property and decidability of satisfiability (without an increase in complexity). The decidability proof uses two-way automata, alternating parity automata of section 5.3 whose modal language is extended with converse modalities [69].

9.2 Least fixpoint logic

Modal logic is a monadic fragment of first-order logic. $L\mu$ is such a fragment of least fixpoint logic, or LFP, obtained by adding fixpoint constructors to first order logic. It is primarily studied in the field of finite model theory; in the realm of infinite models, it is relatively little used, though occasionally used by set theorists as part of the theory of inductive definability. Finite model theorists use various notations, but usually do not use μ and ν, preferring to write LFP/GFP or **lfp/gfp**. We shall stick to a mu-calculus-like notation.

Assume the usual apparatus of first order logic over some structure S. *LFP* is obtained by adding *relation variables* $X, Y, \ldots$ of given arities, and a *least fixpoint operator* μ which forms *relation terms* $\mu X, \vec{x}.\phi$, where $\vec{x}$ is a tuple of arity(X) individual variables, and the relation variable X occurs only positively in ϕ. Assuming a valuation for the other free

variables of ϕ, the semantics of $\mu X, \vec{x}.\phi$ is the least fixpoint of the map $S^n \to S^n$, where n is the arity of X, given by $T \mapsto \{\vec{x} : \phi[X := T]\}$.

LFP has the following properties (refer to a textbook such as [19] for proofs, and for details of results mentioned in this section without citations):

- On finite models with a built-in linear order, LFP captures polynomial time, which makes it useful for complexity theorists. (A logic L captures a complexity class C if every set in C can be defined by a formula of L, and conversely every L-definable set is in C.)
- On finite models, the fixpoint alternation hierarchy collapses, so that any LFP property can be expressed with a single fixpoint; provided that the arity of relation symbols is not bounded. If the arity is bounded, then the fixpoint hierarchy does not collapse.
- LFP does not have the finite model property.
- Satisfiability is undecidable.

LFP retains a fundamental semantic theorem which can be presented as a model-checking game as in section 4.4. The game is now played on an arena of formulae $\phi[s_1, \ldots, s_n]$ with elements s_i of the model for individual variables. The initial position is the starting closed formula ϕ_0 in positive normal form. $\forall$ is responsible for making a move from a position $(\phi \wedge \psi)[s_1, \ldots, s_n]$, where the available choices are $\{\phi[s_1, \ldots, s_n], \psi[s_1, \ldots, s_n]\}$, and from a position $\forall x_{n+1}.\phi[s_1, \ldots, s_n]$, where the available choices are the set $\{\phi[s_1, \ldots, s_n, s] \mid s \in \mathfrak{S}\}$. $\exists$ is responsible for $\vee$ and existential quantification. Final positions are of the form $P[s_1, \ldots, s_n]$ and $\neg P[s_1, \ldots, s_n]$. $\exists$ wins such a position if it is true. Again, $\exists$ wins an infinite play if the outermost fixed point variable Y that occurs infinitely often in the play is a ν-variable. $\exists$ has a history-free winning strategy iff the initial formula is true of the structure.

9.3 Finite variable fixpoint logics

One of the topics studied in finite model theory is the *finite variable* fragments of FOL. These are the fragments FOL^k where the number of distinct variable names in a formula is restricted to a finite value k. Ordinary modal logic is obviously embeddable in FOL^2; there are several features of modal logic that are generalizable in some sense to FOL^2; and by adding certain operators to modal logic, one can regain FOL^2, albeit less succinctly [45]. Moreover, FOL^2 is reasonably tractable, and the decidability of modal logic follows from the decidability of FOL^2, which in turn follows from the fact that, like modal logic and $L\mu$, it enjoys the finite model property.

It is therefore natural to wonder if the good properties of modal mu-calculus might be explained by considering the finite variable fragments of LFP.

However, in a well-known paper 'Why is modal logic so robustly decidable?' [68], Vardi analysed the relationship between modal logic and FOL^2 more carefully, and argued that it does not adequately explain the good properties of modal logic. Furthermore, when one passes to the fixpoint version, it is even more inadequate: for example, although $L\mu$ is decidable, LFP^2 (and $L\mu^2$) is not decidable.

It appears, then, that finite variable fixpoint logics have little to say about $L\mu$. So what are the more useful related logics?

9.4 Guarded fragments

In [68], Vardi argued that the *tree model property* is responsible for the good behaviour of $L\mu$, and CTL. FOL^2 does not have this property. However, it turns out that there are fragments of FOL which do retain the tree model property or some suitable generalization of it. The discovery of these fragments needed a new perception of the characteristic features of modal logic seen as a fragment of FOL.

The fact that modal logic lies in FOL^2 is obvious. Somewhat less obvious is another property of the FO translations of modal logic formulae: *guardedness*. A FO quantification is *guarded* if it has the form $\forall \vec{y}.\alpha(\vec{x}, \vec{y}) \Rightarrow \phi(\vec{x}, \vec{y})$ or $\exists \vec{y}.\alpha(\vec{x}, \vec{y}) \wedge \phi(\vec{x}, \vec{y})$, where $\alpha(\ldots)$ is an atomic formula (i.e. α is a relation symbol or the equality symbol), and $\vec{x}$ includes all the free variables of ϕ. That is, when a quantified variable is introduced, its values must be connected by some relation to the values of the other variables mentioned in the formula. In the case of modal logic, the guards are the edge relations.

Guardedness was proposed by Andréka, van Benthem and Németi [2] as a better explanation of the robust decidability of modal logic. The guarded fragment GF of first-order logic has many of the nice properties of modal logic, for example

- GF is decidable.
- GF has the finite model property.
- GF has the appropriate generalization of the tree model property, namely that if a formula has a model, it has a model of 'bounded tree-width'. (Tree width is a graph-theoretic definition which measures how far a graph is from being a tree.)
- GF-equivalence can be characterized by a *guarded bisimulation*, as modal equivalence is characterized by bisimulation.

Grädel and Walukiewicz [29] studied the guarded fragment GFP of LFP. The syntactic formation rule for fixed points is: if $\phi(Y, \vec{x})$ is a guarded formula, Y occurs positively and not in the guards and all free variables of $\phi(Y, \vec{x})$ are contained in $\vec{x}$ then $\mu Y(\vec{x}).\phi(Y, \vec{x})$ is a formula of the guarded fragment of LFP. This fragment retains the tree model property but not the finite model property, making it a better meta-language for $L\mu$ than LFP^2. An interesting first result concerned the complexity: satisfiability for GFP is 2EXPTIME-complete. Grädel had earlier shown [27] that GF itself has 2EXPTIME satisfiability, so this is a situation where adding fixpoints does not increase complexity - a surprising result. However, it turns out that this depends on the unbounded *width* of formulae - the number of free variables in subformulae. If the width is bounded, then satisfiability drops to EXPTIME-complete, which agrees with that of $L\mu$. The decidability proof uses two-way alternating parity automata.

9.5 Inflationary mu-calculus

In finite model theory, as well as to some extent in classical definability theory, extensions of LFP have been studied which relax the requirement for the body of a fixpoint operator to be monotone. One such is *inflationary fixpoint logic* (IFP). In IFP, the semantics of the fixpoint operator (usually written **ifp** in the finite model theory literature, but here

written μ_{I}) is modified. Rather than being defined as a fixpoint, it is defined in terms of approximants; and then at each approximant, the previous approximant is unioned in:

$$\|\mu_{\mathrm{I}} Z^{\alpha}.\phi\|^{\mathfrak{T}}_{\mathfrak{V}} = Z^{<\alpha} \cup \|\phi\|^{\mathfrak{T}}_{\mathfrak{V}[Z:=Z^{<\alpha}]}$$

On finite structures, IFP and LFP have long been known to be equi-expressive, and recently Kreutzer showed [39] that indeed they are equi-expressive on arbitrary structures. In [18] Dawar, Grädel and Kreutzer define inflationary modal mu-calculus, by using the above definition for fixpoints, and show that it is more powerful than $L\mu$, and complex in many ways. It does not have the finite model property, and it can express non-regular properties. Satisfiability is undecidable and even non-arithmetic, since it is possible to interpret arithmetic, by using the height of nodes in a well-founded tree as numbers. On the class of finite models, the increased power results in a model-checking complexity of PSPACE.

BIBLIOGRAPHY

[1] H. R. Andersen, *Verification of Temporal Properties of Concurrent Systems*, DAIMI PB – 445, Computer Science Dept, Aarhus University, (1993).

[2] H. Andréka, J. van Benthem and I. Németi, Modal languages and bounded fragments of predicate logic. *J. Philos. Logic* 27, 217–274 (1996).

[3] A. Arnold, The μ-calculus alternation hierarchy is strict on binary trees, *Theoretical Informatics and Applications* 33(4/5) 329–340 (1999).

[4] A. Arnold and D. Niwinski, Fixed point characterization of Büchi automata on infinite trees, *J. Inf. Process. Cybern.* EIK 26 451–459 (1990).

[5] A. Arnold and D. Niwinski, *Rudiments of μ-Calculus*, Studies in Logic, Vol. 146, North-Holland (2001).

[6] G. Bhat and R. Cleaveland, Efficient model checking via the equational mu-calculus, *Proc. LICS '96* 304–312 (1996).

[7] J. C. Bradfield, *Verifying Temporal Properties of Systems*, Birkhäuser, Boston (1991).

[8] J. C. Bradfield, The modal mu-calculus alternation hierarchy is strict, *Theor. Comput. Sci.* 195 133–153 (1997).

[9] J. C. Bradfield, Simplifying the modal mu-calculus alternation hierarchy, *Proc. STACS 98* LNCS 1373 39–49 (1998).

[10] J. C. Bradfield, Alternation on the binary tree, *Proc. 1st Int. Workshop on Fixpoints in Computer Science (FICS '98)* (ed. Z. Ésik), Masaryk Univ. Brno, 22–24 (1998).

[11] J. C. Bradfield, Fixpoints in arithmetic, transition systems and trees. *Theoretical Informatics and Applications*, 33(4/5) 341–356 (1999).

[12] J. C. Bradfield and C. P. Stirling, Local model checking for infinite state spaces, *Theoret. Comput. Sci.* 96 157–174, (1992).

[13] J. R. Büchi, On a decision method in restricted second order arithmetic. *Logic, Methodology and Philosophy of Science*, Proc. 1960 congress, Stanford Univ. Press, Stanford, CA, 1–11 (1962).

[14] E. M. Clarke and E. A. Emerson, Design and synthesis of synchronization skeletons using branching time temporal logic. LNCS 131 52–71 (1981).

[15] E. M. Clarke, E. A. Emerson and A. P. Sistla, Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. on Programming Languages and Systems* 8 244–263 (1986).

[16] G. d'Agostino and M. Hollenberg, Logical questions concerning the μ-calculus: interpolation, Lyndon and Łoś–Tarski. *J. Symbolic Logic* 65(1) 310–332 (2000).

[17] M. Dam, CTL* and ECTL* as fragments of the modal mu-calculus. *Theoret. Comput. Sci.* 126 77–96 (1994).

[18] A. Dawar, E. Grädel, and S. Kreutzer, Inflationary fixed points in modal logics, *ACM Trans. on Computational Logic* 5(2) (2004).

[19] H.-D. Ebbinghaus and J. Flum, *Finite Model Theory*, 2nd edition, Springer, 1999.

[20] E. A. Emerson, Temporal and modal logic. *Handbook of Theoretical Computer Science*, Volume B., J. van Leeuwen Editor, Elsevier, 995–1072 (1990).

[21] E. A. Emerson and E. M. Clarke, Characterizing correctness properties of parallel programs using fixpoints. *Procs ICALP '80* LNCS 85 169–181 (1980).
[22] E. A. Emerson and C. S. Jutla, The complexity of tree automata and logics of programs. In *Proc. 29th IEEE FOCS* 328–337 (1988).
[23] E. A. Emerson and C. S. Jutla, Tree automata, mu-calculus and determinacy. In *Proc. 32nd IEEE FOCS* 368–377 (1991).
[24] E. A. Emerson and C.-L. Lei, Efficient model checking in fragments of the propositional mu-calculus, in: *Proc. 1st IEEE LICS* 267–278 (1986).
[25] M. J. Fischer and R. E. Ladner, Propositional dynamic logic of regular programs. *J. Computer and System Science* 18 194–211 (1979).
[26] R. Floyd, Assigning meanings to programs. In J. T. Schwartz, *Mathematical Aspects of Computer Science* 19–32. American Mathematical Society, Providence (1967).
[27] E. Grädel, On the restraining power of guards. *J. Symbol. Logic* 64, 1719–1742 (1999).
[28] E. Grädel, W. Thomas and T. Wilke (Eds.), *Automata, Logics, and Infinite Games*, LNCS 2500 (2002).
[29] E. Grädel and I. Walukiewicz, Guarded fixed point logic, *Proc. 14th LICS* 45 (1999).
[30] Y. Gurevich and L. Harrington, Trees, automata and games. *Procs 14th ACM STOC* 60–65 (1982).
[31] M. Hennessy and R. Milner, On observing nondeterminism and concurrency, *Procs ICALP '80* LNCS 85 295–309 (1980).
[32] D. Janin and I. Walukiewicz, Automata for the μ-calculus and related results. *Proc. MFCS '95* LNCS 969 552–562 (1995).
[33] D. Janin and I. Walukiewicz, On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. *Proc. CONCUR '96* LNCS 1119 263–277 (1996).
[34] R. Kaivola, *Using Automata to Characterise Fixpoint Temporal Logics.* Ph.D. thesis, University of Edinburgh (1997).
[35] D. Kozen, Results on the propositional mu-calculus. *Theoret. Comput. Sci.* 27 333–354 (1983).
[36] D. Kozen, A finite model theorem for the propositional μ-calculus. *Studia Logica* 47 233–241 (1988).
[37] D. Kozen and R. Parikh, An elementary proof of the completeness of PDL. *Theoret. Comput. Sci.* 14 113–118 (1981).
[38] D. Kozen and R. Parikh, A decision procedure for the propositional μ-calculus, LNCS 164 313–325 (1984).
[39] S. Kreutzer, Expressive equivalence of least and inflationary fixed-point logic, *Annals of Pure and Applied Logic* 130(1–3) 61–78 (2004).
[40] O. Kupferman, M. Vardi and P. Wolper, An automata-theoretic approach to branching-time model checking, *J. ACM* 42(2) 312–360 (2000).
[41] R. Küsters and T. Wilke, Deciding the first level of the μ-calculus alternation hierarchy, *Proc. FSTTCS 2002*, LNCS 2556, 241–252 (2002).
[42] L. Lamport, Proving the correctness of multiprocess programs. *IEEE Trans. Software Eng.* 2 125-143 (1977).
[43] G. Lenzi, A hierarchy theorem for the mu-calculus. *Proc. ICALP '96* LNCS 1099 87–109 (1996).
[44] X. Liu, C.R. Ramakrishnan, and S.A. Smolka, Fully local and efficient evaluation of alternating fixed points. *Proc. TACAS '98* LNCS 1384 5–19 (1998).
[45] C. Lutz, U. Sattler and F. Wolter, Modal logic and the two-variable fragment, *Proc. CSL'01* (LNCS 2142), 247– (2001).
[46] A. Mader, *Verification of Modal Properties Using Boolean Equation Systems.* PhD Thesis, Technical University of Munich (1997).
[47] L. Maksimova, Absence of interpolation and of Beth's property in temporal logics with 'the next' operation. *Siberian Mathematical Journal* 32(6) 109–113 (1991).
[48] Z. Manna and A. Pnueli, Formalization of properties of recursively defined functions. *Proc. ACM STOC* 201–210 (1969).
[49] Z. Manna and A. Pnueli, How to cook a temporal proof system for your pet language. *Proc. 10th ACM POPL* 141–154 (1983).
[50] R. Milner, *A Calculus of Communicating Systems.* LNCS 92 (1980).
[51] R. Milner, *Communication and Concurrency.* Prentice-Hall (1989).
[52] A. W. Mostowski, Regular expressions for infinite trees and a standard form of automata, *Computation Theory*, LNCS 208 157–168 (1984).
[53] D. Niwiński, On fixed point clones, *Proc. ICALP '86* LNCS 226 464–473 (1986).
[54] M. Otto, Bisimulation-invariant Ptime and higher-dimensional mu-calculus, *Theoret. Comput. Sci.* 224 237–266 (1999).
[55] M. Otto, Eliminating recursion in the mu-calculus, *Proc. STACS '99*, LNCS 1563 531–540 (1999).
[56] Park, D. Fixpoint induction and proofs of program properties. *Machine Intelligence*, 5, 59-78, Edinburgh University Press (1969).
[57] V. Pratt, Semantical considerations of Floyd–Hoare logic. *Proc. 16th IEEE FOCS* 109–121 (1976).

[58] V. Pratt, A near-optimal method for reasoning about action. *Journal of Comput. and System Sciences* 20 231-254 (1980).
[59] V. Pratt, A decidable mu-calculus, *Proc. 22nd IEEE FOCS*, 421-427 (1982).
[60] M. O. Rabin, Decidability of second-order theories and automata on infinite trees. *Transactions of American Mathematical Society*, 141, 1-35 (1969).
[61] C. Stirling, Local model checking games. *Proc. Concur '95* LNCS 962 1–11 (1995).
[62] C. Stirling and D. Walker, CCS, liveness, and local model checking in the linear time mu-calculus. *Proc. First International Workshop on Automatic Verification Methods for Finite State Systems*, LNCS 407 166–178. (1990).
[63] C. Stirling and D. Walker, Local model checking in the modal mu-calculus. *Theor. Comput. Sci.* 89 161–177. (1991).
[64] R. S. Streett and E. A. Emerson, An automata theoretic decision procedure for the propositional mu-calculus. *Information and Computation* 81 (1989) 249–264.
[65] R. Streett, Propositional dynamic logic of looping and converse. *Proc. 13th ACM STOC* 375–383 (1981).
[66] W. Thomas, Languages, Automata, and Logic, in: *Handbook of Formal Language Theory* (G. Rozenberg, A. Salomaa, Eds.), Vol. III, Springer-Verlag , New York, pp. 389-455 (1998).
[67] M. Y. Vardi, A temporal fixpoint calculus. *Proc. 15th POPL*, 250–259 (1988).
[68] M. Y. Vardi, Why is modal logic so robustly decidable?, in *Descriptive Complexity and Finite Models*, N. Immerman and Ph. Kolaitis, eds., 1997, American Mathematical Society.
[69] M. Y. Vardi, Reasoning about the past with two-way automata, *Proc. ICALP '98* LNCS 1443 628–641 (1998).
[70] M. Y. Vardi and P. Wolper, Yet another process logic. *Logics of Programs* LNCS 164 501–512 (1983).
[71] I. Walukiewicz, Completeness of Kozen's axiomatisation of the propositional μ-calculus. *Information and Computation* 157, 142–182 (2000).

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

13

DESCRIPTION LOGIC

Franz Baader and Carsten Lutz

1 Introduction . . . 757
2 Basic Definitions and Connection to Modal Logic . . . 759
  2.1 Concept Languages . . . 759
  2.2 Terminological Formalisms . . . 763
  2.3 Assertional Formalisms . . . 766
3 Standard Description Logic Inferences . . . 767
  3.1 Terminological Reasoning . . . 767
  3.2 Assertional Reasoning . . . 770
  3.3 Compound Inference Problems . . . 771
4 Sub-Boolean Description Logics . . . 773
  4.1 Reasoning with concept descriptions in sub-Boolean DLs . . . 775
  4.2 TBox reasoning in sub-Boolean DLs . . . 782
  4.3 ABox reasoning in sub-Boolean DLs . . . 786
  4.4 Bi-simulation characterizations of sub-Boolean DLs . . . 787
5 Non-Standard Inferences . . . 788
  5.1 Motivation . . . 789
  5.2 Least common subsumers and most specific concepts . . . 791
  5.3 Matching and unification . . . 793
  5.4 Rewriting and approximation . . . 798
6 Non-Standard Expressivity . . . 801
  6.1 Concrete Domains . . . 801
  6.2 Role Value Maps . . . 808

1 INTRODUCTION

Description logics (DLs) [12] are a family of knowledge representation languages which can be used to represent the terminological knowledge of an application domain in a structured and formally well-understood way. The name *description logics* is motivated by the fact that, on the one hand, the important notions of the domain are described by *concept descriptions*, i.e., expressions that are built from atomic concepts (unary predicates) and atomic roles (binary predicates) using the concept and role constructors provided by the particular DL. For example, the concept of "a man that is married to a doctor, and has only happy children" can be expressed using the concept description

$$\mathsf{Man} \sqcap \exists \mathsf{married}.\mathsf{Doctor} \sqcap \forall \mathsf{child}.\mathsf{Happy}.$$

On the other hand, DLs differ from their predecessors in that they are equipped with a formal, *logic*-based semantics, which can, e.g., be given by a translation into first-order predicate logic. For example, the above concept description can be translated into the following first-order formula (with one free variable x):

$$\mathsf{Man}(x) \wedge \exists y(\mathsf{married}(x,y) \wedge \mathsf{Doctor}(y)) \wedge \forall y(\mathsf{child}(x,y) \rightarrow \mathsf{Happy}(y)).$$

The motivation for introducing the early predecessors of DLs, such as semantic networks and frames [133, 125], actually was to develop means of representation that are closer to the way humans represent knowledge than a representation in formal logics, like first-order predicate logic. Minsky [125] even combined his introduction of the frame idea with a general rejection of logic as an appropriate formalism for representing knowledge. However, once people tried to equip these "formalisms" with a formal semantics, it turned out that they can be seen as syntactic variants of (subclasses of) first-order predicate logic [83, 144].

The immediate precursors of DLs, Brachman's structured inheritance networks [42], were an attempt to define a formalism that allows for a structured representation of knowledge in the spirit of semantics networks and frames, but nevertheless is equipped with a formal semantics. The original description logics used in systems that implemented these ideas in the 1980ies [45, 132, 124, 123] turned out to correspond to rather inexpressive and somewhat unusual subclasses of first-order predicate logic. On the one hand, none of them was propositionally closed since they did not allow for disjunction or negation. On the other hand, they were equipped with certain other complex constructors (like number restrictions and role-value-maps), which, though expressible in first-order predicate logic, are not considered as atomic constructors there. For example, the number restriction $(\geqslant 5\,\mathsf{child})$ describes people having at least five children, and the role-value-map $\mathsf{child} \circ \mathsf{friend} \subseteq \mathsf{know}$ describes people that know all their children's friends.

The main inference problem to be solved in description logics is the subsumption problem, i.e., deciding whether one concept is a subconcept of another one. The early DL systems cited above employed so-called structural subsumption algorithms, which first normalise the concept descriptions, and then recursively compare the syntactic structure of the normalised descriptions. These algorithms are usually very efficient (polynomial), but they have the disadvantage that they are complete only for rather inexpressive DLs, i.e., for more expressive DLs they cannot detect all the existing subsumption relationships. To overcome this problem, Schmidt-Schauß and Smolka [143] made DLs into "real" logics by introducing negation. Their main motivation for this was that they wanted to reduce the subsumption problem to the satisfiability problem. They introduced a basic propositionally closed DL, which they called $\mathcal{ALC}$, developed a tableau-like algorithm for satisfiability in $\mathcal{ALC}$, and showed that the subsumption and satisfiability problem in $\mathcal{ALC}$ are PSPACE-complete.

A reader of the Handbook of Modal Logic who followed us so far may rightfully ask: *And what has all this to do with Modal Logic?* The answer was given by Schild, who noticed that $\mathcal{ALC}$ is just a syntactic variant of multi-modal K, i.e., the basic modal logic of Kripke frames with several accessibility relations (and thus several pairs of box- and diamond operators). In fact, the translations of $\mathcal{ALC}$ and of K into first-order predicate logic yield exactly the same class of first-order formulae. This connection between DLs and modal logic was used by Schild and others (see, e.g., [139, 140, 54, 55]) to transfer decidability and complexity results from modal logic to DLs, but also to extend these

results to logics with other DL constructors. At the same time, tableau-based algorithms were developed for more and more expressive DLs (see [30] for an overview), and highly-optimized implementations of these algorithms [92] turned out to behave quite well on artificial benchmarks from modal logic [131] and also in practice [78].

Though there is a very close connection between DLs and modal logics (MLs), the underlying intuition as well as the intended applications differ significantly. As a consequence, the focus of research in DL and in modal logic also differs. While mentioning the similarities, this chapter will focus on topics that are specific for DLs.

Section 2 formally introduces syntax and semantics of the basic DL $\mathcal{ALC}$, and shows its relationship to multi-modal K. It then introduces additional DL constructors, and describes their ML counterparts. In addition to these constructors, DLs provide their users with a terminological formalism, which (in its simplest form) allows to introduce names for complex concepts, and an assertional formalism, which allows to state facts about specific individuals/objects. Though these components are usually not available in ML, there are some connections to things known in ML (such as nominals, the universal modality, fixpoint operators, etc.).

In *Section 3*, we introduce the standard inference problems in description logics, show how they can be reduced to each other, and how they relate to inference problems in ML. The standard way of solving these problems in propositionally closed DLs is using tableau-based algorithms. Since these algorithms are treated in other chapters, we only give some references to the relevant chapters.

Section 4 considers DLs that are not propositionally closed, and where consequently subsumption cannot be reduced to satisfiability. We review the known complexity results for such DLs, and then describe (complete) structural subsumption algorithms for some of them. In addition, we mention bi-simulation characterizations of the corresponding ML fragments.

Section 5 is concerned with so-called non-standard inferences in DLs, like computing the least common subsumer and the most specific concept, and rewriting, unification, and matching of concepts. These inferences have been introduced with the goal of supporting the user when building and maintaining large DL knowledge bases. With the exception of unification, none of them have been investigated in ML.

Finally, *Section 6* introduces means of expressiveness that do not have immediate ML counterparts.

2 BASIC DEFINITIONS AND CONNECTION TO MODAL LOGIC

In this section, we introduce the basic components of description logics: concept languages, terminological formalisms, and assertional formalisms.

2.1 Concept Languages

We first define the basic propositionally closed concept language $\mathcal{ALC}$ introduced by Schmidt-Schauß and Smolka [143], and then describe a number of natural extensions that are important for many applications and offered by modern DL reasoners. Assume that a countably infinite supply of *concept names*, usually denoted A and B, and of *role names*, usually denoted r and s, are available. *Concept descriptions* in $\mathcal{ALC}$ are formed

Name	Syntax	Semantics
top concept	$\top$	$\Delta^{\mathcal{I}}$
bottom concept	$\bot$	$\emptyset$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
value restriction	$\forall r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \forall e.(d,e) \in r^{\mathcal{I}} \rightarrow e \in C^{\mathcal{I}}\}$
existential restriction	$\exists r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e.(d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$
transitive role	r	$r^{\mathcal{I}}$ transitive
inverse role	r^{-}	$\{(d,e) \mid (e,d) \in r^{\mathcal{I}}\}$
nominal	I	$I^{\mathcal{I}}$ singleton
qualifying number restrictions	$\leqslant n\ r\ C$ $\geqslant n\ r\ C$	$\{d \in \Delta^{\mathcal{I}} \mid \#\{(d,e) \in r^{\mathcal{I}} \mid d \in C^{\mathcal{I}}\} \leq n\}$ $\{d \in \Delta^{\mathcal{I}} \mid \#\{(d,e) \in r^{\mathcal{I}} \mid e \in C^{\mathcal{I}}\} \geq n\}$
number restrictions	$\leqslant n\ r$ $\geqslant n\ r$	$\{d \in \Delta^{\mathcal{I}} \mid \#\{(d,e) \in r^{\mathcal{I}}\} \leq n\}$ $\{d \in \Delta^{\mathcal{I}} \mid \#\{(d,e) \in r^{\mathcal{I}}\} \geq n\}$

Figure 1. Semantics of concept and role constructors

according to the following syntax rule:

$$C, D \longrightarrow A \mid \top \mid \bot \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall r.C \mid \exists r.C$$

where A ranges over concept names and r ranges over role names. In examples, we will usually use uppercase names for concept names and lowercase names for role names, thus obtaining $\mathcal{ALC}$ concept descriptions such as the one given in the introduction:

$$\mathsf{Man} \sqcap \exists \mathsf{married}.\mathsf{Doctor} \sqcap \forall \mathsf{child}.\mathsf{Happy}.$$

The semantics of $\mathcal{ALC}$ is based on *interpretations*, i.e., pairs $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a non-empty set (the *domain* of $\mathcal{I}$), and $\cdot^{\mathcal{I}}$ is the *interpretation function*, assigning to each concept name A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to each role name r a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function is inductively extended to concept descriptions as shown in (the upper part of) Figure 1, which also lists the names that we use for $\mathcal{ALC}$ constructors. An interpretation $\mathcal{I}$ is a *model* of a concept description C if $C^{\mathcal{I}} \neq \emptyset$. In the following, we will sometimes call concept languages "description logics", ignoring further ingredients to DLs such as the terminological formalism.

As first observed by Schild [138], $\mathcal{ALC}$ is a notational variant of the multi-modal logic K. Syntactically, concept names can simply be viewed as propositional variables and role names can be viewed as names for accessibility relations. Then, interpretations of $\mathcal{ALC}$ are obviously just Kripke structures with $\Delta^{\mathcal{I}}$ the set of worlds and $\cdot^{\mathcal{I}}$ providing both the accessibility relations and the valuation of the propositional variables. With this reading, the value restriction $\forall r.C$ becomes a box operator $\Box_r C$ referring to the accessibility relation denoted by r, and $\exists r.C$ becomes a diamond operator $\Diamond_r C$. This connection is also witnessed by the usual translation of $\mathcal{ALC}$ to first-order predicate logic [37, 38], which is identical to the standard translation for modal logic presented in Chapter 1.

The concept language $\mathcal{ALC}$ is only one member of a large family of concept languages. These languages can be obtained from $\mathcal{ALC}$ by disallowing certain constructors (thus obtaining the *sub-Boolean* description logics discussed in Section 4) and/or adding various combinations of additional constructors. Such additional constructors can be concept constructors, or they can be role constructors allowing to construct compound *role descriptions* to be used in place of role names. We now discuss several additional constructors that are related to expressive means common in modal logic. Constructors that have been considered in the context of description logics, but lack a modal counterpart will be discussed in more depth in Section 6.

When the connection between DLs and modal logic was discovered, one of it's first uses was to transfer results from propositional dynamic logic (PDL) to description logics [138, 139, 54]. The description logic counterpart of PDL is called $\mathcal{ALC}_{reg}$, which stands for "$\mathcal{ALC}$ with *reg*ular expressions on roles" [3, 138]. $\mathcal{ALC}_{reg}$ extends $\mathcal{ALC}$ by allowing compound role descriptions inside value restrictions and existential restrictions. Such role descriptions are built using the binary constructors for union ("$\sqcup$") and composition (";") and a unary constructor for reflexive-transitive closure ("$\cdot^*$"). The semantics is given in the straightforward way by interpreting the constructors using the corresponding relational operations. For example, the additional constructors could be used in the concept description

$$\mathsf{Man} \sqcap \exists \mathsf{child}.\mathsf{Human} \sqcap \forall(\mathsf{child};\mathsf{child}^*).\mathsf{Happy}$$

where $(\mathsf{child};\mathsf{child}^*)$ describes the transitive closure of the role child, i.e., the $\mathsf{descendant}$ relation. The work on $\mathcal{ALC}_{reg}$ has led to several variants and extensions whose expressive power goes beyond that of PDL [54, 56, 57]. However, many of today's most used concept languages do not include the role constructors of $\mathcal{ALC}_{reg}$. The main reason is that applications demand an implementation of description logic reasoning, and the presence of the reflexive-transitive closure constructor makes obtaining efficient implementations much harder.

Another important family of description logics is obtained by considering fragments of the concept language $\mathcal{SHOIQ}$ [94, 98, 95], which extends $\mathcal{ALC}$ with several expressive means that are discussed in detail below.[1] The importance of $\mathcal{SHOIQ}$ stems from the fact that it and its fragments are used in two of the most influential application areas of description logics: reasoning about conceptual database models [52] and reasoning in the semantic web [19]. In the latter application, the fragment $\mathcal{SHOIN}$ roughly corresponds to the ontology language OWL-DL [93], which was recommended by the W3C as the standard web ontology language. The fragment $\mathcal{SHIQ}$ is the concept language supported by modern description logic systems such as FaCT and RACER [91, 79]. A tableau algorithm for full $\mathcal{SHOIQ}$ was introduced in [97], and optimized implementations of this algorithm are under development.

Compared to $\mathcal{ALC}$, the additional expressive means provided by $\mathcal{SHOIQ}$ are transitive roles, role hierarchies, inverse roles, qualifying number restrictions, and nominals. With the exception of role hierarchies, the formal semantics of these extensions can be found in the lower part of Figure 1. Below, we discuss each means of expressiveness in more detail. Before that, a remark on the naming scheme used to describe fragments of $\mathcal{SHOIQ}$ is in order. To avoid long sequences of letters, the abbreviation $\mathcal{S}$ was introduced for $\mathcal{ALC}$

[1] The naming of description logics is historically grown, and there are several naming schema in use; see the Appendix of [12].

Symbol	Syntax	$\mathcal{SHIQ}$	$\mathcal{SHOIQ}$	$\mathcal{SHIN}$	$\mathcal{SHOIN}$
$\mathcal{H}$	$r \sqsubseteq s$	x	x	x	x
$\mathcal{I}$	r^-	x	x	x	x
$\mathcal{N}$	$(\leqslant n\, r), (\geqslant n\, r)$	x	x	x	x
$\mathcal{Q}$	$(\leqslant n\ r\ C), (\geqslant n\ r\ C)$	x	x		
$\mathcal{O}$	I		x		x

Figure 2. Some members of the $\mathcal{S}$ family of DLs.

with transitive roles. The additional presence of role hierarchies is indicated by the letter $\mathcal{H}$, of inverse roles by $\mathcal{I}$, of (qualifying) number restrictions by $\mathcal{N}$ ($\mathcal{Q}$), and of nominals by $\mathcal{O}$ (see Figure 2).

Transitive roles. $\mathcal{ALC}$ can be extended with transitive roles by adding a new sort of role names whose interpretation is required to be transitive [135]. The resulting description logic is a notational variant of the fusion of multi-modal K and multi-modal K4. One of the most important uses of transitive roles is for the representation of knowledge about parts and wholes by means of a transitive role part-of [136].

Inverse roles extend $\mathcal{ALC}$ with a unary role constructor $\cdot^-$. Roles of the form r^- correspond to "backwards modalities" as known from temporal logic and converse PDL [154]. They allow, for example, to define the converse parent of the relation child, and the converse has-part of part-of.

Role hierarchies are not a part of the concept language, but rather "external" to it [91]. Formally, a role hierarchy is a finite set of inclusion statement $r \sqsubseteq s$ with r and s role descriptions. Intuitively, the presence of a role hierarchy puts constraints on the class of accepted interpretations: if $r \sqsubseteq s$ is in the hierarchy, then we only accept interpretations in which $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$. Thus, role hierarchies are much closer in spirit to TBoxes (see below) than to concept or role constructors. The connection between role hierarchies and modal logics will be discussed in a more general context in Section 6.2.

Nominals are an additional sort of concept names that are required to be interpreted as singleton sets. The name has been adopted from modal logics, where nominals appear e.g. in the context of hybrid logic, c.f. Chapter 14 and [1, 74]. There are several natural concepts, such as Pope, that require nominals for an adequate modelling. In description logics, nominals sometimes occur in the form of two concept constructors called "one of" and "fills", see [44] for more details.

Qualifying number restrictions. Corresponding to graded modalities in modal logic [70, 71, 153], qualifying number restrictions allow to put *counting* constraints on the number of domain elements that are related via a certain role and belong to a certain concept [88]. This constructor allows, e.g., the formulation of concepts such as Father $\sqcap$ ($\leqslant$ 1 child Female) describing fathers that have at most one daughter (but arbitrarily many sons). Number restrictions also appear in a non-qualifying variant $\leqslant n\ r$ and $\geqslant n\ r$, in which the third argument implicitly is the top concept. They are a very important means of expressivity that appeared already in early description logic systems such as KL-ONE [45]. In the case of $\mathcal{SHOIQ}$ and its fragments, number restrictions are usually restricted to *simple* roles, i.e. roles having no transitive subroles according to the role hierarchy. If this syntactic restriction is not adopted, reasoning in $\mathcal{SHOIQ}$ is undecidable [98].

For the sake of brevity, the list of concept and role constructors given above is not

$$
\begin{array}{rclcrcl}
\mathsf{Woman} & \equiv & \mathsf{Person} \sqcap \mathsf{Female} & \qquad & \mathsf{Man} & \equiv & \mathsf{Person} \sqcap \neg\mathsf{Woman} \\
\mathsf{Mother} & \equiv & \mathsf{Woman} \sqcap \exists \mathsf{child}.\mathsf{Person} & & \mathsf{Father} & \equiv & \mathsf{Man} \sqcap \exists \mathsf{child}.\mathsf{Person} \\
\mathsf{Parent} & \equiv & \mathsf{Mother} \sqcup \mathsf{Father} & & & &
\end{array}
$$

Figure 3. An example TBox formulated in $\mathcal{ALC}$.

exhaustive. For example, $\mathcal{ALC}$ has also been extended with Boolean role constructors, which corresponds to going from multi-modal K to Boolean modal logic [101, 122, 121].

2.2 Terminological Formalisms

The concept language is only one part of description logics. To capture the terminological knowledge of application domains in a structured way, it is not sufficient to formulate single concept descriptions. Additionally, we must be able to organize and interrelate multiple concept descriptions in a suitable way. This is achieved through the terminological formalism. Just like concept languages, terminological formalisms come in several flavors. One of the most fundamental variants is the following: a *TBox (terminological box)* $\mathcal{T}$ is a finite set of *concept definitions*

$$A \equiv C$$

with A a concept name and C a concept description, such that no concept name appears on the left-hand side of two different concept definitions in $\mathcal{T}$. An example of a TBox formulated in $\mathcal{ALC}$ is displayed in Figure 3. A concept name is called a *defined concept* if it appears on the left-hand side of a concept definition, and a *primitive concept* otherwise.

When defining the semantics, we face the difficulty of treating terminological cycles which may occur in the kind of TBoxes considered here. We say that a concept name A *directly uses* a concept name B w.r.t. a TBox $\mathcal{T}$ if there is a concept definition $A \equiv C \in \mathcal{T}$ with B occurring C. Let *uses* be the transitive closure of directly uses. Then a TBox $\mathcal{T}$ contains a *terminological cycle* if there is a concept name that uses itself w.r.t. $\mathcal{T}$; otherwise $\mathcal{T}$ is called *acyclic*. For example, the TBox displayed in Figure 3 is acyclic, whereas the following concept definition induces a terminological cycle (Adam and Eve are nominals):

$$\mathsf{Human} \equiv \mathsf{Adam} \sqcup \mathsf{Eve} \sqcup \exists \mathsf{parent}.\mathsf{Human}.$$

In the following, we sometimes call the general form of TBoxes introduced above *cyclic TBoxes* to distinguish them from acyclic ones. However, this does not imply that the TBoxes in question *necessarily* contains a terminological cycle.

For acyclic TBoxes, the natural semantics is *descriptive semantics*: an interpretation $\mathcal{I}$ *satisfies* a concept definition $A \equiv C$ if $A^{\mathcal{I}} = C^{\mathcal{I}}$, and $\mathcal{I}$ is a *model* of the TBox $\mathcal{T}$ if it satisfies all concept definitions in $\mathcal{T}$. Intuitively, acyclic TBoxes merely state that defined concepts are abbreviations for certain compound concept descriptions. These compound concepts can be made explicit by *expanding* the acyclic TBox $\mathcal{T}$: exhaustively replace all concept names A on the left-hand side of concept definitions $A \equiv C$ by their defining concept descriptions C. After this expansion, the compound concept abbreviated by a

defined concept can simply be read off from the corresponding concept definition. For example, the defined concept Father in Figure 3 abbreviates the compound concept

$$\mathsf{Person} \sqcap \neg(\mathsf{Person} \sqcap \mathsf{Female}) \sqcap \exists \mathsf{child}.\mathsf{Person}.$$

A *primitive interpretation* for a TBox $\mathcal{T}$ is an interpretation that interprets only the primitive concept names and role names, but not the defined concepts. A (full) interpretation $\mathcal{I}$ is called an *extension* of a primitive interpretation $\mathcal{J}$ if it agrees with $\mathcal{J}$ on the domain and the interpretation of the primitive concepts and role names. We say that $\mathcal{T}$ is *definitorial* if every primitive interpretation has exactly one extension that is a model of $\mathcal{T}$. Since we can expand them, acyclic TBoxes are clearly definitorial: if $\mathcal{T}$ is an acyclic TBox and $\mathcal{T}' = \{A_1 \equiv C_1, \ldots, A_k \equiv C_k\}$ has been obtained from $\mathcal{T}$ by expansion, then the unique extension of a primitive interpretation $\mathcal{J}$ that is a model of $\mathcal{T}$ is obtained by setting $A_i^{\mathcal{I}} := C_i^{\mathcal{J}}$ for $1 \leq i \leq k$.

If we do not require TBoxes to be acyclic, then TBoxes are no longer definitorial under descriptive semantics. For example, the TBox

$$\mathcal{T} \equiv \{\mathsf{Human} \equiv \forall \mathsf{parent}.\mathsf{Human}\}$$

has no primitive concept, and the primitive interpretation $\mathcal{J}$ with $\Delta^{\mathcal{J}} = \{d\}$ and $\mathsf{parent}^{\mathcal{J}} = \{(d,d)\}$ can be extended to two different models of $\mathcal{T}$. Thus, the above TBox does not provide an unequivocal definition of Human. To obtain definitorial TBoxes in the presence of terminological cycles, two steps are necessary [129]: first, descriptive semantics is changed to a (least/greatest) fixpoint semantics; and second, the syntax of TBoxes is restricted to ensure that least and greatest fixpoints indeed exist. To illustrate why fixpoints are a natural choice for defining TBox semantics, we note that they can be used to characterize models of a TBox in a straightforward way. Let $\mathcal{T}$ be a TBox and $\mathcal{J}$ a primitive interpretation for $\mathcal{T}$. We write $\mathcal{T}(A)$ to denote the concept description C if $A \equiv C \in \mathcal{T}$. With $Ext_{\mathcal{J}}$, we denote the set of all extensions of $\mathcal{J}$. Let $\mathcal{T}_{\mathcal{J}} : Ext_{\mathcal{J}} \to Ext_{\mathcal{J}}$ be the mapping that maps the extension $\mathcal{I}$ of $\mathcal{J}$ to the extension $\mathcal{T}_{\mathcal{J}}(\mathcal{I})$ of $\mathcal{J}$ defined by setting $A^{\mathcal{T}_{\mathcal{J}}(\mathcal{I})} := (\mathcal{T}(A))^{\mathcal{I}}$ for each defined concept A. It is trivial to verify that an interpretation $\mathcal{I}$ is a model of $\mathcal{T}$ if and only if $\mathcal{I}$ is a fixpoint of $\mathcal{T}_{\mathcal{J}}$ with $\mathcal{J}$ the restriction of $\mathcal{I}$ to a primitive interpretation.

To make the TBox formalism definitorial in the presence of terminological cycles, we restrict the set of fixpoints of $\mathcal{T}_{\mathcal{J}}$ that are intended as models. Let $\mathcal{I}$ be a model of $\mathcal{T}$ and $\mathcal{J}$ the restriction of $\mathcal{I}$ to a primitive interpretation. Then $\mathcal{I}$ is a *least fixpoint model* (*greatest fixpoint model*) of $\mathcal{T}$ if $A^{\mathcal{I}} \subseteq A^{\mathcal{I}'}$ ($A^{\mathcal{I}} \supseteq A^{\mathcal{I}'}$) for every defined concept A and every fixpoint $\mathcal{I}'$ of $\mathcal{T}_{\mathcal{J}}$. We obtain the *least fixpoint semantics* (*greatest fixpoint semantics*) by admitting only the least fixpoint models (greatest fixpoint models) of $\mathcal{T}$ as intended models. However, the obtained semantics is still not definitorial, at least not for all TBoxes: let $\mathcal{T} = \{A \equiv \forall r.\neg A\}$ and $\mathcal{J}$ the primitive interpretation with $\Delta^{\mathcal{J}} = \{d\}$ and $r^{\mathcal{J}} = \{(d,d)\}$. Then $\mathcal{J}$ has no extension that is a model of $\mathcal{T}$. The usual way to get around such a problem, as e.g. used in the modal μ-calculus [104], is to adopt a syntactic monotonicity restriction. In the setting of cyclic TBoxes, this restriction can be formulated as follows: a TBox $\mathcal{T}$ is called *monotone* if, on the right-hand side of concept definitions in $\mathcal{T}$, defined concepts appear only under an even number of negations. It is easy to show that, according to least or greatest fixpoint semantics every monotone TBox is definitorial: every primitive interpretation can be *uniquely* extended to a least or greatest fixpoint model of the TBox.

Whether least fixpoint semantics or greatest fixpoint semantics is preferable depends on the concept definition at hand: for the concept definition $\mathsf{Human} \equiv \mathsf{Adam} \sqcup \mathsf{Eve} \sqcup \exists \mathsf{parent}.\mathsf{Human}$ from above, we should use the least fixpoint semantics to avoid that individuals on cyclic or infinite parent-paths have to be Humans. In other cases, greatest fixpoint semantics can be more appropriate: say we want to define top researchers (in a somewhat incestuous way) as researchers who are renowned and collaborate only with top researchers:

$$\mathsf{TopResearcher} \equiv \mathsf{Researcher} \sqcap \mathsf{Renowned} \sqcap \forall \mathsf{collaborates\text{-}with}.\mathsf{TopResearcher}.$$

Then least fixpoint semantics is not convincing since two renowned researchers who collaborate mutually (but not with anybody else) will not be classified as top researchers. In contrast, greatest fixpoint semantics yields the intended models. These two examples illustrate that the most flexible solution is to use a mixed semantics: least fixpoints for some defined concepts, and greatest fixpoints for others [140]. Note that, in the case of an acyclic TBox, least fixpoint semantics, greatest fixpoint semantics, and descriptive semantics coincide in the sense that they admit exactly the same models.

It is also possible to use descriptive semantics for cyclic TBoxes. As discussed above, this implies that TBoxes will no longer be definitorial. While this is inappropriate if the goal is to *define* concepts, it poses no problem if we view TBoxes simply as formulating *constraints* on the intended models. This view of TBoxes, which is rather natural in a number of applications, leads to the idea of *general concept inclusion axioms (GCIs)*. A GCI is an expression of the form

$$C \sqsubseteq D,$$

where both C and D are (possibly compound) concept descriptions. An interpretation $\mathcal{I}$ *satisfies* the GCI $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. When working with GCIs as constraints on models, no syntactic restrictions such as unique left-hand sides, acyclicity, or monotonicity needs to be adopted. For example, we could use a GCI to state that all persons having an uncle who is a father also have a cousin:[2]

$$\mathsf{Person} \sqcap \exists \mathsf{uncle}.\mathsf{Father} \sqsubseteq \exists \mathsf{cousin}.\mathsf{Person}$$

Since the concept definition $A \equiv C$ can be rewritten as the pair of GCIs $A \sqsubseteq C$ and $C \sqsubseteq A$, GCIs strictly generalize acyclic TBoxes as well as cyclic TBoxes with descriptive semantics. It should be noted that GCIs are the terminological formalism that is usually supported by modern description logic systems.

We now discuss the relation between terminological formalisms and modal logic. In the case of descriptive semantics, there is a close relationship to the universal modality: let $\mathcal{T}$ be a set of GCIs and U the universal role, i.e. $U^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ for all interpretations $\mathcal{I}$. Then we can translate $\mathcal{T}$ into a concept $C_{\mathcal{T}}$ by setting

$$C_{\mathcal{T}} := \forall U. \bigsqcap_{D \sqsubseteq E \in \mathcal{T}} \neg D \sqcup E.$$

Then we have the following: if an interpretation $\mathcal{I}$ is a model of $\mathcal{T}$, then $C_{\mathcal{T}}^{\mathcal{I}} := \Delta^{\mathcal{I}}$; and if $C_{\mathcal{T}}^{\mathcal{I}} \neq \emptyset$, then $\mathcal{I}$ is a model of $\mathcal{T}$. We will see in Section 3.1 that this translation can sometimes be used to reduce reasoning with TBoxes to reasoning without TBoxes.

[2] This could be modelled in an even better way using role value maps, c.f. Section 6.

$\mathsf{Logician}(\mathsf{DAVID})$	$\mathsf{supervisor}(\mathsf{DONALD}, \mathsf{VAUGHAN})$
$\mathsf{supervisor}(\mathsf{VAUGHAN}, \mathsf{DAVID})$	$(\mathsf{Man} \sqcap \exists \mathsf{child}.\mathsf{Woman})(\mathsf{DONALD})$

Figure 4. An example ABox formulated in $\mathcal{ALC}$.

In a weaker sense, we can also do the converse translation, i.e. simulate the universal modality using GCIs—see Section 2.2.1 of [112] for more details.

In the case of fixpoint semantics, Schild [140] observed that there is a direct correspondence between TBoxes and (an alternation-free fragment of) Vardi and Wolper's version of the propositional μ-calculus [155]. In contrast to the standard μ-calculus as proposed by Kozen [104], this variant provides for multiple fixpoints that correspond to constructing fixpoints for all defined concepts of a TBox simultaneously.

Finally, there is an intimate connection between our notion of definitorial TBoxes and the Beth definability property as known from modal logic [72]. Roughly, a description logic has the Beth definability property if and only if every TBox that is definitorial under descriptive semantics is equivalent to an acyclic TBox (see also [29]).

2.3 Assertional Formalisms

Apart from the concept language and the terminological formalism, there is one more important ingredient to description logics. This is the *assertional formalism*, which allows to describe (a snapshot of) the world by means of individuals populating the world, conceptual memberships of individuals, and roles relating individuals. The combination of a TBox and an ABox is commonly called a *knowledge base*. Assume that a countably infinite supply of individual names, usually denoted by a, b, c, is available. An *ABox (assertional box)* is a finite set of assertions of the form

$$\begin{array}{rl} C(a) & \quad \text{(concept assertion)} \\ r(a,b) & \quad \text{(role assertion)} \end{array}$$

where a and b are individual names, C is a concept description, and r is a role description. An example of an ABox is given in Figure 4. We us all-uppercase words to denote concrete individual names.

To assign a semantics to ABoxes, we have to extend interpretations to individual names: interpretations $\mathcal{I}$ are now required to map, additionally, every individual name a to a domain element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. Usually, the *unique name assumption (UNA)* is adopted, which requires that different individual names are mapped to distinct domain elements, i.e., $a \neq b$ implies $a^{\mathcal{I}} \neq b^{\mathcal{I}}$. The interpretation $\mathcal{I}$ *satisfies* the concept assertion $C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$, and it *satisfies* the role assertion $r(a,b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. An interpretation is a *model* of an ABox $\mathcal{A}$ if it satisfies all assertions in $\mathcal{A}$. Often, we are interested in models of an ABox $\mathcal{A}$ *w.r.t. a TBox* $\mathcal{T}$, i.e. common models of $\mathcal{A}$ and $\mathcal{T}$.

There is an obvious connection between ABoxes and nominals which provides a link between ABoxes and modal logic: if a concept language providing for nominals, conjunction, and existential restrictions is used, then we can simulate an ABox $\mathcal{A}$ using the

concept description

$$C_{\mathcal{A}} := \bigsqcap_{D(a)\in\mathcal{A}} \exists u.(a \sqcap D) \sqcap \bigsqcap_{r(a,b)\in\mathcal{A}} \exists u.(a \sqcap \exists r.b)$$

where u is a role name not used in $\mathcal{A}$, and we assume that, for each individual name, there exists a nominal of the same name.[3] This is a simulation in the sense that every model for $C_{\mathcal{A}}$ is a model for $\mathcal{A}$, and every model $\mathcal{I}$ for $\mathcal{A}$ can be extended to a model for $C_{\mathcal{A}}$ by setting $u^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. However, nominals are strictly more expressive than ABoxes. For example, this is reflected by the complexity of reasoning in $\mathcal{ALCI}$, the extension of $\mathcal{ALC}$ with inverse roles: while reasoning in $\mathcal{ALCI}$ with ABoxes but without TBoxes is PSPACE-complete, reasoning in $\mathcal{ALCI}$ extended with nominals is EXPTIME-complete.[4] The latter is due to the possibility of defining "spy-points" in $\mathcal{ALCI}$ with nominals (see Chapter 14), which is not possible using ABoxes.

3 STANDARD DESCRIPTION LOGIC INFERENCES

Reasoning has always been a major emphasis of description logic research. The main purpose of reasoning in DLs is to explicate knowledge that is contained only implicitly in a given concept description, TBox, or ABox. This inferencing capability can be used by applications to infer new knowledge when needed, and it helps knowledge engineers to construct and structure complex knowledge bases. In this section, we introduce the inference problems for description logics that have direct counterparts in modal logic. Because these inference problems have played an important rôle since the very beginnings of description logic, they are often referred to as "standard inference problems"—in contrast to the more recent "non-standard inference problems" that are discussed in Section 5. We also give a brief survey of the most important results and techniques concerning the decidability and computational complexity of the standard inference problems. In doing so, we concentrate on description logics that have close counterparts in modal logics and defer the treatment of logics that are less common from the modal logic perspective to Section 6. Our discussion of results and techniques will be brief as these or very similar issues are covered in more detail in other chapters of this handbook.

3.1 Terminological Reasoning

The inference problems introduced here operate on concept descriptions and TBoxes, without reference to ABoxes. The basic such inference problems are the following:

Satisfiability. A concept description C is *satisfiable* with respect to a TBox $\mathcal{T}$ if there exists a common model of C and $\mathcal{T}$.

Subsumption. A concept description C is *subsumed* by a concept description D with respect to a TBox $\mathcal{T}$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{T}$ (written $C \sqsubseteq_{\mathcal{T}} D$).

In both cases, we simply drop the reference "with respect to $\mathcal{T}$" (and the index $\mathcal{T}$ from $C \sqsubseteq_{\mathcal{T}} D$) if we are interested in reasoning w.r.t. the empty TBox. In this case, we also talk about reasoning *with concept descriptions.* Intuitively, satisfiability is important to

[3]The additional role can be omitted if the "$@_a$" operator of hybrid logic is used, c.f. Chapter 14.
[4]With "reasoning" we refer to ABox consistency, c.f. Section 3.2.

(automatically) verify whether a concept description makes sense from a logical perspective, i.e., whether it is contradictory in itself or to a given TBox. Satisfiability also plays an important rôle because many other inference problems can be reduced to it. Subsumption can be used to check whether a concept D is more general than a concept C, i.e., whether each instance of C also is an instance of D. For example, the concept name Parent is subsumed by the concept description $\mathsf{Man} \sqcup \mathsf{Woman}$ w.r.t. the TBox shown in Figure 3: by the semantics, every Parent is also a Man or a Woman. As we will discuss in more detail later, subsumption defines a hierarchy of the concept names occurring in a TBox w.r.t. their generality. There are some additional terminological inference problems such as the equivalence of concept descriptions: C and D are *equivalent* with respect to a TBox $\mathcal{T}$ (written $C \equiv_{\mathcal{T}} D$) iff $C^{\mathcal{I}} = D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$. We will not consider such additional inference problems in this chapter since they can clearly be reduced to subsumption and satisfiability in a trivial way.

There is a straightforward connection between satisfiability and subsumption: if a concept language provides for negation and conjunction, we can polynomially reduce subsumption to unsatisfiability: $C \sqsubseteq_{\mathcal{T}} D$ if and only $C \sqcap \neg D$ is unsatisfiable w.r.t. $\mathcal{T}$. We can also do the converse reduction if the concept language provides for (or can express) the bottom concept: C is satisfiable w.r.t. $\mathcal{T}$ if and only if $C \not\sqsubseteq_{\mathcal{T}} \bot$. Because of this close connection, many description logic systems concentrate on providing algorithms for solving satisfiability, and treat subsumption by means of the above reduction.[5] Another important group of reductions is concerned with reducing satisfiability with respect to TBoxes to satisfiability w.r.t. the empty TBox. Whether such a reduction can be done depends on the concept language and the chosen TBox formalism. Here we discuss the two most important cases.

Eliminating acyclic TBoxes. As already mentioned in Section 2.2, acyclic TBoxes merely define abbreviations for compound concept descriptions. This suggests the following reduction: to decide whether a concept description C is satisfiable w.r.t. the acyclic TBox $\mathcal{T}$, first expand $\mathcal{T}$ (c.f. Section 2.2), then replace all defined concepts in C according to their definition in the expansion of $\mathcal{T}$, and finally decide satisfiability of the resulting concept description without reference to a TBox. As observed by Nebel [128], this reduction may yield an exponential blowup even for the concept language $\mathcal{FL}_0$ that only provides for the concept constructors conjunction and value restriction. For example, expanding the following TBox of size $\mathcal{O}(n)$ yields a TBox of size 2^n:

$$
\begin{array}{rcl}
C_1 & \equiv & \forall r_1.C_0 \sqcap \forall r_2.C_0 \\
 & \vdots & \\
C_n & \equiv & \forall r_1.C_{n-1} \sqcap \forall r_2.C_{n-1}
\end{array}
$$

This exponential blowup can sometimes be avoided by devising satisfiability algorithms that explicitly take acyclic TBoxes into account. For example, satisfiability of $\mathcal{ALC}$ concept descriptions w.r.t. acyclic TBoxes is PSpace-complete, and without TBoxes this problem is of exactly the same complexity [142, 110]. However this is not always the case: in Sections 4 and 6, we will discuss DLs for which reasoning w.r.t. acyclic TBoxes is considerably more difficult than reasoning without them.

Eliminating GCIs. In several expressive description logics, it is possible to reduce satisfiability w.r.t. GCIs to satisfiability without reference to GCIs. Two examples are the

[5] An important exception are sub-Boolean DLs that do not provide for general negation; c.f. Section 4.

description logics $\mathcal{ALC}_{reg}$ and $\mathcal{SHOIQ}$. It is not difficult to prove that, in $\mathcal{ALC}_{reg}$, a concept description C is satisfiable w.r.t. $\mathcal{T}$ if, and only if, $C \sqcap \forall(r_1 \sqcup \cdots \sqcup r_k)^*.C_{\mathcal{T}}$ is satisfiable, where $r_1, \ldots, r_k$ are the role names used in C and $\mathcal{T}$, and $C_{\mathcal{T}}$ is defined at the end of Section 2.2. Similarly, it has been observed in [94] that, in $\mathcal{SHOIQ}$, a concept description C is satisfiable w.r.t. a TBox $\mathcal{T}$ and a role hierarchy $\mathcal{H}$[6] if and only if $C \sqcap \forall r.C_{\mathcal{T}}$ is satisfiable w.r.t. the role hierarchy $\mathcal{H} \cup \{r_1 \sqsubseteq r, \ldots, r_k \sqsubseteq r\}$, where $r_1, \ldots, r_k$ are the role names used in C and $\mathcal{T}$, and r is a transitive role not occurring in C and $\mathcal{T}$. Reductions like the ones sketched above are often called *internalizations* of TBoxes, and have first been proposed in [11]. However, implemented reasoning systems usually treat GCIs in an explicit way for efficiency reasons [90, 99].

We now give a brief survey of the results and techniques for terminological reasoning. The main driving force behind the research on DL reasoning is the following trade-off between expressivity and computational complexity: on the one hand, non-trivial applications require a high expressivity of the concept language and of the terminological and assertional formalism; on the other hand, applications need an implementation of DL inference algorithms in an actual knowledge representation system that exhibits an acceptable run-time behavior on "realistic" inputs, i.e. on inputs that stem from an application and have not been artificially crafted to make reasoning hard.

It is generally agreed upon that an implemented DL system should be based on algorithms that are sound, complete, and terminating, i.e., decidability of the relevant inference problems is indispensable. Fortunately, satisfiability and subsumption is indeed decidable for almost all possible combinations of concept language and TBox formalism that we have introduced up to this point, including $\mathcal{ALC}$ with cyclic TBoxes and fixpoint semantics [140], $\mathcal{ALC}_{\mathrm{reg}}$ with GCIs [54], and $\mathcal{SHOIQ}$ with GCIs [150].[7] However, decidability of reasoning is usually only a necessary, but not a sufficient condition for the usefulness of a description logic. Additionally, it is important that the computational complexity of reasoning is within acceptable bounds, and that there exist *practical* reasoning algorithms, i.e. algorithms that have the potential of being implemented in a system that behaves well on realistic inputs as demanded above.

The general opinion on the (worst-case) complexity that is acceptable has changed dramatically over time. Historically, the early times of DL research have been concentrating on identifying formalisms for which reasoning is tractable, i.e. can be performed in polynomial time.[8] Obviously, demanding tractability means that we cannot include all Boolean operators in the concept language, and thus are in the realm of sub-Boolean DLs. The complexity of satisfiability and subsumption in this family of DLs is laid out in detail in Section 4, ranging from tractable to ExpTime-complete depending on the choice of constructors and TBox formalism. Around 1990, the $\mathcal{KRIS}$ system showed that tableau algorithms for satisfiability and subsumption in $\mathcal{ALC}$ w.r.t. acyclic TBoxes, two PSpace-complete inference problems, can be implemented in a system with acceptable run-time behavior on realistic inputs [17]. A step towards even more expressive DLs has been made around 1997 by Ian Horrocks and his FaCT system, which originally implemented satisfiability and subsumption for an ExpTime-complete fragment of

[6] C is satisfiable w.r.t. $\mathcal{T}$ and $\mathcal{H}$ if there is a model $\mathcal{I}$ of C and $\mathcal{T}$ with $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$ for all $r \sqsubseteq s \in \mathcal{H}$.

[7] An exception is satisfiability of $\mathcal{SHOIQ}$-concepts w.r.t. TBoxes with least fixpoint semantics, which was shown to be undecidable by Bonatti [35].

[8] Curiously, it was found later that reasoning in the description logic supported by the very first description logic system KL-ONE is undecidable [141]—c.f. Section 5.

$\mathcal{SHOIQ}$ with GCIs. The complexity of most description logics extending $\mathcal{ALC}$ is between PSpace-complete and NExpTime-complete. Some important landmarks are the following:

- satisfiability of $\mathcal{ALC}$ concept descriptions without reference to TBoxes is PSpace-complete [142]; this also holds in the presence of acyclic TBoxes [110];
- satisfiability of $\mathcal{ALC}$ concept descriptions w.r.t. cyclic TBoxes or GCIs is ExpTime-complete; this holds for fixpoint semantics as well as for descriptive semantics [138, 140];
- satisfiability of $\mathcal{SHOIQ}$ concept descriptions w.r.t. GCIs is NExpTime-complete, with the lower bound applying to all extensions of $\mathcal{ALC}$ that provide for (qualifying or non-qualifying) number restrictions, inverse roles, and nominals [150].

All these bounds transfer to subsumption with the exception of the last one, where NExpTime-completeness of satisfiability flips to co-NExpTime-completeness of subsumption. It should be mentioned that the exact meaning of an "acceptable run-time behavior" of course also depends on the concrete application at hand. As argued e.g. in [51], there are applications that require "real" tractability and therefore research in tractable DLs is an ongoing endeavour [51, 46, 10]. Since our survey of the complexity of reasoning in DLs is by no means exhaustive, we refer the interested reader to [61] for more information on the complexity of DLs.

The issue of practicability is not only related to computational complexity, but also to the techniques that are used to obtain decision procedures for DL reasoning. A large number of such techniques have been proposed and investigated. For sub-Boolean DLs, so-called "structural algorithms" play the most important role, and we describe them in detail in Section 4. For $\mathcal{ALC}$ and its many extensions, the following approaches are most important: tableau algorithms [30], reduction techniques [54], automata-based approaches [122, 50], and resolution calculi [101, 100]. With respect to practicability, tableau algorithms are the most successful approach so far: they proved to be amenable to a number of powerful optimization techniques (see Chapter 4 and [92]), and highly-optimized implementations of tableau algorithms in DL systems have performed extraordinarily well in system comparisons. As a result, nowadays almost all state-of-the-art DL reasoners, such as FaCT and RACER [91, 79], are based on tableau algorithms.

From the perspective of modal logic, satisfiability of concept descriptions clearly corresponds to standard formula satisfiability, whereas a subsumption $C \sqsubseteq D$ corresponds to the validity of the implication $C \rightarrow D$. For this reason, the discussion of tableau- and resolution-based algorithms for modal logics provided in Chapter 4 of this handbook applies to description logics as well, and we omit further details.

3.2 Assertional Reasoning

The inference problems discussed in this section operate on knowledge bases, i.e. on pairs $(\mathcal{A}, \mathcal{T})$ with $\mathcal{A}$ an ABox and $\mathcal{T}$ a TBox. The fundamental inference problems are the following:

Consistency. An ABox $\mathcal{A}$ is *consistent* w.r.t. a TBox $\mathcal{T}$ if there exists a common model of $\mathcal{A}$ and $\mathcal{T}$.

Instance Checking. An individual name a in an ABox $\mathcal{A}$ is an *instance* of a concept description C w.r.t. a TBox $\mathcal{T}$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ (denoted with $\mathcal{A} \models_{\mathcal{T}} C(a)$).

Consistency is the ABox-analogue of satisfiability, i.e., it can be used to check whether a given knowledge base is contradictory. The purpose of instance checking is also obvious: it is used to derive concept memberships of individuals that are not stated explicitly. For example, the individual DONALD in the ABox shown in Figure 4 is an instance of Father w.r.t. the TBox in Figure 3.

As in the previous section, there is a close connection between the two fundamental inference problems if certain Boolean constructors are available. First, consistency can be polynomially reduced to (non-)instance checking if the bottom concept is available: an ABox $\mathcal{A}$ is consistent w.r.t. a TBox $\mathcal{T}$ if and only if $\mathcal{A} \not\models_{\mathcal{T}} \bot(a)$ with a an arbitrary individual name. And second, we can do the converse reduction if full negation is available: $\mathcal{A} \models_{\mathcal{T}} C(a)$ if and only if $\mathcal{A} \cup \{\neg C(a)\}$ is inconsistent w.r.t. $\mathcal{T}$. It is sometimes also possible to eliminate acyclic TBoxes and GCIs as discussed in the previous section. In the presence of nominals, it is possible to polynomially reduce consistency (and thus also instance checking) to the satisfiability of concept descriptions using the simulation sketched at the end of Section 2.3.

The techniques used to devise decision procedures for consistency and instance checking are essentially the same as those employed for concept satisfiability and subsumption. In the case of tableau algorithms, there are two approaches for reasoning with ABoxes: first, ABox consistency can sometimes be reduced to concept satisfiability using the *pre-completion* technique described in [87]; and second, tableau algorithms can be extended to treat ABoxes in a direct way (see e.g. [86, 15, 77]). Regarding practicability, it should be noted that some optimization techniques fail or become significantly more complex in the presence of ABoxes.

Concerning the decidability and computational complexity of assertional reasoning, one should distinguish sub-Boolean DLs from $\mathcal{ALC}$ and its extensions. In the sub-Boolean case, there are some description logics for which instance checking is harder than concept subsumption (see Section 4.3). If all Boolean constructors are available, the complexity of instance checking coincides with the complexity of subsumption for all such description logics investigated so far. For example, instance checking in $\mathcal{ALC}$ ABoxes without reference to TBoxes is known to be PSPACE-complete [18], and instance checking in $\mathcal{ALC}$ ABoxes w.r.t. GCIs is EXPTIME-complete [54]—just as the corresponding cases of subsumption.

3.3 Compound Inference Problems

Some of the most important inference problems in DLs are of a compound nature in the sense that, in principle, they can be reduced to multiple invocations of the more basic inference problems mentioned above. However, when the goal is to achieve an efficient implementation, it is vital to consider compound inferences as first-class citizens [13]. Here we discuss the three most important such problems.

Classification. Given a TBox $\mathcal{T}$, compute the restriction of the subsumption relation "$\sqsubseteq_{\mathcal{T}}$" to the set of concept names used in $\mathcal{T}$.

Realization. Given an ABox $\mathcal{A}$, a TBox $\mathcal{T}$, and an individual name a, compute the set $R_{\mathcal{A},\mathcal{T}}(a)$ of those concept names A that are used in $\mathcal{T}$, satisfy $\mathcal{A} \models_{\mathcal{T}} A(a)$, and are

minimal with this property w.r.t. the subsumption relation "$\sqsubseteq_{\mathcal{T}}$".

Retrieval. Given an ABox $\mathcal{A}$, a TBox $\mathcal{T}$, and a concept C, compute the set $I_{\mathcal{A},\mathcal{T}}(C)$ of individual names a used in $\mathcal{A}$ and satisfying $\mathcal{A} \models_{\mathcal{T}} C(a)$.

Compound inferences are a very important interface to description logics reasoners and are offered by almost all systems. The purpose of classification is to construct a hierarchy of concept names w.r.t. their generality, with more general concepts higher up in the hierarchy: B is above A if and only if $A \sqsubseteq_{\mathcal{T}} B$. Such a hierarchy can then be presented to a knowledge engineer for browsing and structuring the TBox. Realization also facilitates browsing and understanding of the knowledge base, and is a precursor to certain operations on knowledge bases that presuppose knowledge of the concept memberships of individuals. The main use of retrieval is database-like querying of description logic knowledge bases: in some applications, it is natural to define ABoxes with a huge number of individual names, and to query such ABoxes like a database with deductive capabilities [78].

By definition, compound inferences can be reduced to more basic inference problems. For classification, we may simply check whether $A \sqsubseteq_{\mathcal{T}} B$ for all concept names A, B used in $\mathcal{T}$. In the case of realization, we can obviously just use multiple invocations of instance checking and subsumption. Similarly, multiple instance checks suffice to get a naïve implementation of retrieval. However, basic inferences such as subsumption and instance checking are potentially very costly, and thus it is vital for DL reasoners to replace these "brute force" methods of compound inferences by more subtle approaches.

To illustrate how compound inferences can be implemented in a more efficient way, we exemplarily consider classification. Here, the aim is to minimize the number of subsumption tests, of which the naïve approach performs n^2 many with n the number of concept names in $\mathcal{T}$. The common strategies for achieving this minimization are described and evaluated by Baader et al. in [13]. Although Baader et al. restrict themselves to acyclic TBoxes, the proposed strategies can also be used for cyclic ones. In general, two kinds of optimizations can be distinguished. Firstly, classification can be conceived as an abstract combinatorial problem on partial orders: compute a complete representation of a partial ordering by making as few as possible comparisons. This quite general problem is also considered in non-DL contexts, see e.g. [69]. Secondly, we can take into account the structure of concept descriptions to reveal obvious subsumption relationships and to control the order in which concepts are added to the hierarchy. In the following, we assume that the restriction of "$\sqsubseteq_{\mathcal{T}}$" to the concepts names of $\mathcal{T}$ is represented as a Hasse diagram, i.e. as a directed acyclic graph (DAG) such that

- nodes are sets of concept names that are pair-wise equivalent w.r.t. $\mathcal{T}$;
- two nodes S_1, S_2 are connected by an edge if every $A_2 \in S_2$ is a *direct subsumer* of every $A_1 \in S_1$, i.e., we have (i) $A_1 \sqsubseteq_{\mathcal{T}} A_2$ and (ii) $A_1 \sqsubseteq_{\mathcal{T}} B \sqsubseteq_{\mathcal{T}} A_2$ implies $B \equiv_{\mathcal{T}} A_1$ or $B \equiv_{\mathcal{T}} A_2$ for all concept names B in $\mathcal{T}$.

To this diagram, we henceforth refer as the *(concept) hierarchy*. We assume that the hierarchy always contains a top node whose label includes $\top$, and a bottom node whose label includes $\bot$.[9] In DL TBoxes originating from applications, the concept hierarchy is usually not too deep, i.e., the represented order has short chains and long antichains.

[9] In case of unsatisfiable TBoxes, the top node and bottom node coincide.

One way to compute the concept hierarchy with only few subsumption tests is to use an incremental algorithm [13]: we start with a hierarchy containing only $\top$ and $\bot$, and then repeatedly place additional concept names at the appropriate position in the (growing) hierarchy. The placing of a new concept name A consists of two phases: a *top search* phase computing the direct subsumers of A that are already contained in the hierarchy, and a *bottom search* phase computing the set of all concept names that are already contained in the hierarchy, and of which A is a direct subsumer. Obviously, knowledge of these two sets allows us to place A appropriately. Due to the transitivity of the subsumption relation, the top search phase is best implemented as a top down search, whereas a bottom up approach is appropriate for the bottom search phase. Additionally, failed tests can be propagated down the hierarchy in the top search phase: if $A \not\sqsubseteq_{\mathcal{T}} B$ and B' is below B in the hierarchy (implying $B' \sqsubseteq_{\mathcal{T}} B$), then it follows immediately that $A \not\sqsubseteq_{\mathcal{T}} B'$. Analogously to propagation in the top search phase, successful tests can be propagated up the hierarchy in the bottom search phase. Finally, it is possible to use information gained in the top search phase to speed up the bottom search phase, and vice versa (see [13] for details).

Using the structure of concepts, we can additionally avoid subsumption tests in a straightforward way: if we find a concept definition $A \equiv C$ with C a conjunction having as one of its conjuncts a concept name B, then $A \sqsubseteq_{\mathcal{T}} B$ holds trivially. In this case, B is a *told subsumer* of A. Of course, if B is a defined concept, it can have told subsumers as well, and these (and their told subsumers, etc.) can also be viewed as told subsumers of A. The information about the told subsumers can be propagated down the hierarchy before starting the top search phase. To take full advantage of this idea, it is advisable to classify concepts in *definition order*. This means that a concept is not classified until all of its told subsumers are classified.

These optimizations typically reduce the number of necessary subsumption tests to a small fraction of n^2 (see [13] for details). Most techniques sketched here can be used in the same or a slightly modified form if sets of GCIs are used instead of TBoxes. Of course, it is (at least) equally important to optimize the subsumption test itself. More on this issue can be found in Chapter 4.

4 SUB-BOOLEAN DESCRIPTION LOGICS

As mentioned in the introduction, the DLs used in the first DL systems did not allow for all Boolean operators. Usually, these DLs provided for conjunction, value-restriction, and number restriction, and some other special constructors, but existential restriction, disjunction and full negation were not available. In some of these formalisms, disjointness statements between concept names or atomic negation (i.e., negation restricted to concept names) were allowed.

This restriction to sub-Boolean logics was, on the one hand, due to the origins of DLs. These formalisms were not primarily seen as logics (where the inclusion of at least proposition logic is natural), but as knowledge representation formalisms in the spirit of semantic networks and frames, though equipped with a formal semantics. Graph-based formalisms like semantic networks usually favor a conjunctive point of view since conjunction corresponds to just drawing several things in the same picture, whereas expressing disjunction and negation would require special conventions (like drawing a box around the parts that are negated, as in conceptual graphs [147]), which easily

destroy the readability of such graphical representations.

On the other hand, the restriction to sub-Boolean DLs was motivated by the goal of designing representation formalisms with tractable (i.e., polynomial-time decidable) inference problems, which would be precluded by the presence of all Boolean operators. The first paper addressing the trade-off between expressiveness and tractability of reasoning in the context of DL was [109], where it was shown that a seemingly minor extension of the description language can make the subsumption problem intractable. This work triggered an extensive investigation of the borderline between tractability and intractability of reasoning in sub-Boolean DLs [127, 145, 63, 62, 40, 65].[10] In *Section 4.1*, we give a brief review of these results. We also sketch in more detail polynomial-time subsumption algorithms for the DL $\mathcal{FL}_0$ (which allows for conjunction and value restriction only) and some of its extensions. The results mentioned until now were all restricted to extensions of $\mathcal{FL}_0$. The reason was that until the late 1990ies, both conjunction and value restriction were assumed to be indispensable for a DL. For conjunction this indeed appears to be the case since one usually wants to require several properties simultaneously when defining a concept. In order to obtain more than just a fragment of propositional logic, one also needs at least one constructor involving roles. However, instead of value restrictions one could also use existential restrictions. In fact, there are large DL-based medical terminologies [134, 148] that employ existential restrictions rather than value restrictions. The complexity of reasoning in DLs extending $\mathcal{EL}$, which allows for conjunction, existential restriction, and the top-concept, is less well-investigated than for extensions of $\mathcal{FL}_0$. We will briefly review the results in [84], where the complexity of the satisfiability problem in all fragments of $\mathcal{ALC}$, including ones that do not extend $\mathcal{FL}_0$, is investigated. In addition, we sketch a polynomial-time subsumption algorithms for $\mathcal{EL}$.

All the complexity results mentioned until now are concerned with satisfiability and subsumption of concept descriptions. If one considers reasoning w.r.t. a TBox, then the complexity may increase drastically, even for acyclic TBoxes, which do not increase the expressive power. The first such result is due to Nebel [128], who showed that the subsumption problem w.r.t. acyclic TBoxes is coNP-complete for the DL $\mathcal{FL}_0$. Recall that subsumption of $\mathcal{FL}_0$ concept descriptions is polynomial. If one allows for cyclic TBoxes, then subsumption in $\mathcal{FL}_0$ becomes PSpace-complete, and in the presence of GCIs it becomes even ExpTime-complete. In contrast, the subsumption problem in $\mathcal{EL}$ remains polynomial in the presence of acyclic or cyclic TBoxes and GCIs. These results for reasoning w.r.t. TBoxes in sub-Boolean DLs will be described in more detail in *Section 4.2*.

Not only TBox reasoning, but also ABox reasoning, may be harder than reasoning with concept descriptions. *Section 4.3* gives an example, due to A. Schaerf [137], of a sub-Boolean DL where the instance problem is harder than the subsumption problem.

Finally, *Section 4.4* reviews bisimulation characterizations for various sub-Boolean DLs due to de Rijke and Kurtonina [105], which can be used to characterize the expressive power of these DLs.

[10] We have not included [64] in this list since the polynomiality results for subsumption claimed there turned out to be incorrect (see [66] for details).

Symbol	Syntax	$\mathcal{ALN}$	$\mathcal{ALE}$	$\mathcal{ALU}$	$\mathcal{ALUN}$	$\mathcal{ALEN}$	$\mathcal{ALC}$	$\mathcal{ALCN}$
$\mathcal{E}$	$\exists r.C$		x			x	x	x
$\mathcal{U}$	$C \sqcup D$			x	x		x	x
$\mathcal{N}$	$(\leqslant n\, r), (\geqslant n\, r)$	x			x	x		x

Figure 5. The $\mathcal{AL}$ family of DLs.

4.1 Reasoning with concept descriptions in sub-Boolean DLs

Donini et al. [65] start their investigation of the complexity of sub-Boolean DLs with the DL $\mathcal{AL}$, whose concept descriptions are formed according to the following syntax rule:

$$C, D \longrightarrow A \mid \top \mid \bot \mid \neg A \mid C \sqcap D \mid \forall r.C \mid \exists r.\top$$

The difference to $\mathcal{ALC}$ is that the application of negation is restricted to concept names (atomic negation) and that in existential restrictions only the top concept may occur (restricted existential quantification).

In the following, we consider the extensions of $\mathcal{AL}$ by subsets of the following set of constructors: (full) existential restriction, number restrictions, and disjunction.[11] This yields the 7 different extensions of $\mathcal{AL}$ shown in Figure 5. Note that adding both existential restriction and disjunction to $\mathcal{AL}$ yields $\mathcal{ALC}$. This is due to the presence of atomic negation in $\mathcal{AL}$. In fact, by using de Morgan's law, the duality of the quantifiers, and the removal of double negation, any $\mathcal{ALC}$ concept description can be transformed into an equivalent one that employs only atomic negation.

Figure 6 gives a complete classification of the DLs belonging to the $\mathcal{AL}$ family regarding the worst-case complexity of subsumption and (un)satisfiability of concept descriptions. Except for the case of $\mathcal{ALEN}$, these results are shown in [65]. PSPACE-hardness of $\mathcal{ALEN}$ was shown by Hemaspaandra [84].

Before trying to explain some of these results, let us first point out that subsumption and unsatisfiability are in general not trivially interreducible in sub-Boolean DLs. We have seen that

$$\begin{aligned} C \text{ is unsatisfiable} \quad &\text{iff} \quad C \sqsubseteq \bot, \\ C \sqsubseteq D \quad &\text{iff} \quad \neg C \sqcap D \text{ is unsatisfiable.} \end{aligned}$$

Since $\bot$ is available in the DLs of the $\mathcal{AL}$ family, unsatisfiability can be reduced in this way to subsumption, and thus subsumption is at least as hard as unsatisfiability, and unsatisfiability is at least as easy as subsumption. However, for a DL strictly below $\mathcal{ALC}$, $\neg C \sqcap D$ usually does not belong to this DL even if C and D do. Nevertheless, the results summarized in Figure 6 show that, for the $\mathcal{AL}$ family, the complexities of unsatisfiability and of subsumption coincide. In general, this need not be the case. For example, the extension $\mathcal{ALNI}$ of $\mathcal{ALN}$ by inverse roles has a polynomial-time (un)satisfiability problem, but the subsumption problem is coNP-hard [66]. In very simple DLs such as $\mathcal{FL}_0$ and $\mathcal{EL}$, satisfiability is even trivial (in contrast to subsumption) since there are no unsatisfiable concepts We exemplarily discuss at least one DL for each of the complexity classes appearing in Figure 6. Satisfiability in intractable DLs is treated in Section 4.1 and subsumption in tractable DLs such as $\mathcal{FL}_0$ and $\mathcal{ALN}$ is discussed in Section 4.1.

[11] Donini et al. [65] actually consider a somewhat larger family of DLs, where also intersection of roles is available as a constructor.

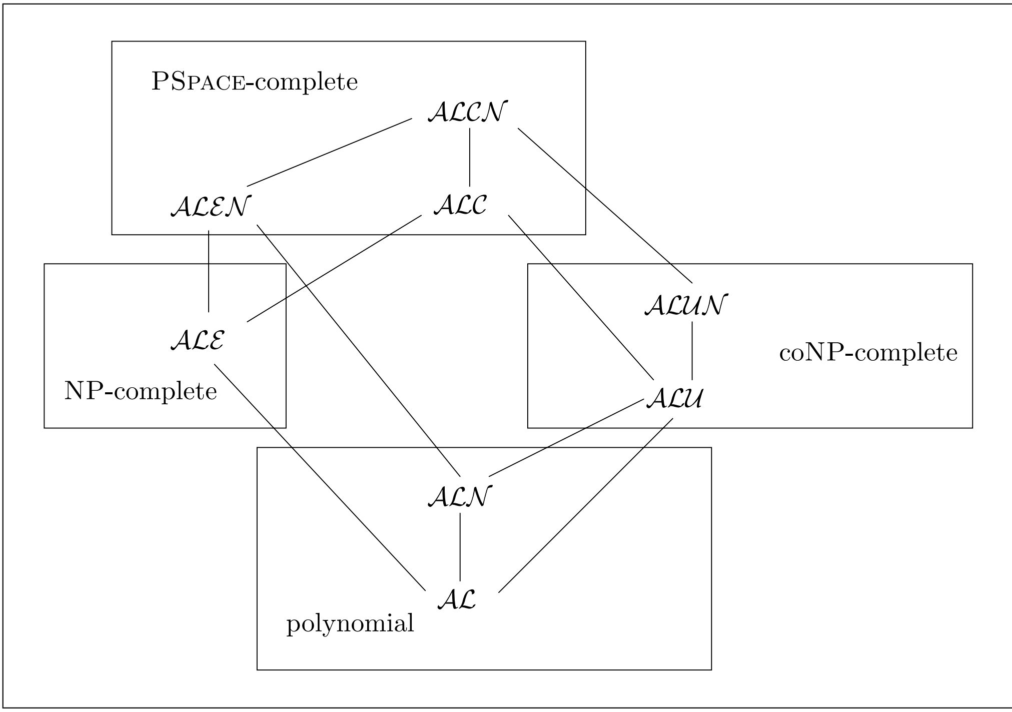


Figure 6. The complexity of unsatisfiability and subsumption for the $\mathcal{AL}$ family of DLs.

In Section 4.1 we review the results from [84] on satisfiability in other sub-Boolean DLs, and in Section 4.1 we sketch a polynomial-time subsumption algorithm for $\mathcal{EL}$ concept description.

(Un)satisfiability in $\mathcal{ALC}$, $\mathcal{ALU}$, and $\mathcal{ALE}$

One way of deciding satisfiability in $\mathcal{ALC}$ is to use a tableau algorithm, as described in more detail in Chapter 4 for the modal logic equivalent K_n of $\mathcal{ALC}$. This algorithm tries to generate a finite, tree-shaped model for a given input concept description C. This tree model is of depth linear in the size of C, but may still be exponentially large due to the branching in the tree model. There are two sources of complexity for this approach: first, the potentially exponential size of the model that must be generated, and second the non-deterministic treatment of disjunction when trying to generate the model. In order to stay within PSpace, one generates one branch of the tree at a time, whereas non-determinism is harmless since it is well-known that NPSpace = PSpace.

If we restrict this approach to $\mathcal{ALU}$, then it is easy to see that the tree models generated by a tableau-based algorithm are of polynomial size. Thus, to decide satisfiability within NP (and thus unsatisfiability within coNP) one can simply guess an interpretation of polynomial size, and check whether it is a model. NP-hardness is trivial since $\mathcal{ALU}$ contains full propositional logic.

In the case of $\mathcal{ALE}$, the model generated by a tableau-based algorithm may still be exponentially large, but the process of generating it is deterministic. In order to check

unsatisfiability within NP (and thus satisfiability within coNP), one can guess one path through the potential model, and then check whether it must satisfy contradictory constraints. To be more precise, instead of trying to generate successors for all existential restrictions, one non-deterministically chooses the ones that actually lead to a contradiction. Since the paths are linear in the size of the input description, this leads to an NP-algorithm for unsatisfiability.

Showing the lower complexity bound for $\mathcal{ALE}$ is less trivial than for $\mathcal{ALU}$. To show that unsatisfiability of concept descriptions in $\mathcal{ALE}$ is NP-complete, we sketch the reduction from set traversal given in [62]. An instance of the *set traversal problem* is given by a finite collection $\mathcal{M} = \{M_1, \ldots, M_m\}$ of finite sets of positive integers. A *set traversal* is a finite set of positive integers N such that $N \cap M_\ell$ is a singleton set for all $\ell, 1 \leq \ell \leq m$. NP-hardness of the existence of a set traversal is an immediate consequence of the fact that monotone ONE-IN-THREE 3SAT [73] is a special case of this problem.

Let $\mathcal{M} = \{M_1, \ldots, M_m\}$ be an instance of the set traversal problem, and assume without loss of generality that the numbers occurring in the sets are the ones from 1 to n for some positive integer n. We define the corresponding $\mathcal{ALE}$ concept description as

$$C_{\mathcal{M}} := C_1 \sqcap \ldots \sqcap C_n \sqcap D,$$

where

$$C_\ell := Q_{1,\ell}r.Q_{2,\ell}r.\cdots Q_{m,\ell}r.\ Q_{1,\ell}r.Q_{2,\ell}r.\cdots Q_{m,\ell}r.\ \top$$

such that

$$Q_{i,\ell} = \begin{cases} \exists & \text{if } \ell \in M_i, \\ \forall & \text{if } \ell \notin M_i, \end{cases}$$

and D is the nesting of $2m$ value restrictions followed by $\bot$, i.e.,

$$D := \underbrace{\forall r.\cdots\forall r.}_{2m \text{ times}} \bot.$$

As an example, consider the two instances

$$\mathcal{M} := \{\{1,3,5\}, \{2,4\}, \{4,5\}\} \qquad \text{and} \qquad \mathcal{M}' := \{\{1,3\}, \{2,4\}, \{4,5\}\}$$

of the set traversal problem. Then the corresponding $\mathcal{ALE}$ concept descriptions look as follows:

$$\begin{array}{ll}
C_1 = \exists\mathbf{r}.\forall r.\forall r.\ \exists\mathbf{r}.\forall r.\forall.r\ \top, & C_1' = \exists\mathbf{r}.\forall r.\forall r.\ \exists\mathbf{r}.\forall r.\forall.r\ \top, \\
C_2 = \forall r.\exists\mathbf{r}.\forall r.\ \forall r.\exists\mathbf{r}.\forall r.\ \top, & C_2' = \forall r.\exists\mathbf{r}.\forall r.\ \forall r.\exists\mathbf{r}.\forall r.\ \top, \\
C_3 = \exists\mathbf{r}.\forall r.\forall r.\ \exists\mathbf{r}.\forall r.\forall r.\ \top, & C_3' = \exists\mathbf{r}.\forall r.\forall r.\ \exists\mathbf{r}.\forall r.\forall r.\ \top, \\
C_4 = \forall r.\exists\mathbf{r}.\exists\mathbf{r}.\ \forall r.\exists\mathbf{r}.\exists\mathbf{r}.\ \top, & C_4' = \forall r.\exists\mathbf{r}.\exists\mathbf{r}.\ \forall r.\exists\mathbf{r}.\exists\mathbf{r}.\ \top, \\
C_5 = \exists\mathbf{r}.\forall r.\exists\mathbf{r}.\ \exists\mathbf{r}.\forall r.\exists\mathbf{r}.\ \top, & C_5' = \forall r.\forall r.\exists\mathbf{r}.\ \forall r.\forall r.\exists\mathbf{r}.\ \top, \\
D = \forall r.\forall r.\forall r.\ \forall r.\forall r.\forall r.\ \bot, & D' = \forall r.\forall r.\forall r.\ \forall r.\forall r.\forall r.\ \bot.
\end{array}$$

One can view the quantifier prefixes as a matrix, where the rows correspond to the elements of the sets, whereas the columns correspond to the sets (written twice). The existential quantifier indicates that the element belongs to the respective set. For example, the first existential quantifier in C_5 expresses that 5 belongs to the first set of $\mathcal{M}$, whereas the universal quantifier at the same position in C_5' says that 5 does not belong to the first set of $\mathcal{M}'$.

In [62], it is shown that $C_{\mathcal{M}}$ is unsatisfiable iff $\mathcal{M}$ has a set traversal. We illustrate the connection between unsatisfiability of $C_{\mathcal{M}}$ and the existence of a set traversal for $\mathcal{M}$ on our example. For $C_{\mathcal{M}}$ to be unsatisfiable, the conjunction of the concept descriptions C_i must enforce an r-path of length $2m$, which then clashes with the $\bot$ in D. The set $\{1,2,5\}$ is a set traversal for $\mathcal{M}'$. This implies that $C'_1 \sqcap C'_2 \sqcap C'_5$ enforces a path of length 6. In fact, C'_1 starts with an existential restriction, and C'_2 and C'_5 with value restrictions. Thus, there is an r-successor of the initial element that must satisfy the conjunction of the three concept description C''_1, C''_2, C''_5 that are respectively obtained from C'_1, C'_2, C'_5 by removing the first quantifier. Now C''_2 starts with an existential quantifier, and the other two with value restrictions. Thus, we can continue with the corresponding r-successor. In general, the properties of a set traversal ensure that each time we have one existential restriction, whereas all the others are value restrictions (and thus the remaining parts of the concept descriptions are propagated to the r-successor required by the existential restriction).

It remains to explain why we have to encode the sets twice. Again, we illustrate this on our example. It is easy to see that the collection $\mathcal{M} := \{\{1,3,5\}, \{2,4\}, \{4,5\}\}$ does not have a set traversal. Nevertheless, if we consider the simpler reduction concept $\widehat{C}_{\mathcal{M}} := \widehat{C}_1 \sqcap \widehat{C}_2 \sqcap \widehat{C}_3 \sqcap \widehat{C}_4 \sqcap \widehat{C}_5 \sqcap \widehat{D}$ where

$$\begin{aligned}
\widehat{C}_1 &= \exists\mathbf{r}.\forall r.\forall .r\ \top, \\
\widehat{C}_2 &= \forall r.\exists\mathbf{r}.\forall r.\ \top, \\
\widehat{C}_3 &= \exists\mathbf{r}.\forall r.\forall r.\ \top, \\
\widehat{C}_4 &= \forall r.\exists\mathbf{r}.\exists\mathbf{r}.\ \top, \\
\widehat{C}_5 &= \exists\mathbf{r}.\forall r.\exists\mathbf{r}.\ \top, \\
\widehat{D} &= \forall r.\forall r.\forall r.\ \bot,
\end{aligned}$$

then $\widehat{C}_{\mathcal{M}}$ is unsatisfiable. In fact, $\widehat{C}_4 \sqcap \widehat{C}_5$ enforce a path of length 3. The corresponding set $\{4,5\}$ is not a set traversal since its intersection with $M_3 = \{4,5\}$ is not a singleton. For the shorter reduction concept $\widehat{C}_{\mathcal{M}}$, the fact that $\widehat{C}_4$ and $\widehat{C}_5$ have an existential restriction in the same row is irrelevant. However, for the correct longer reduction concept $C_{\mathcal{M}}$ this means that, for one of the two, the remaining part is missing in the second round.

Subsumption in $\mathcal{FL}_0$ and $\mathcal{ALN}$

Subsumption in $\mathcal{ALN}$ can be decided in polynomial time using a *structural subsumption algorithm*, i.e., an algorithm that normalizes the descriptions to be tested for subsumption, and then compares the syntactic structure of the normal forms. For simplicity, we first explain the ideas underlying this approach for the small DL $\mathcal{FL}_0$, which allows for conjunction ($C \sqcap D$) and value restriction ($\forall r.C$) only. Subsequently, we show how the bottom concept ($\bot$), atomic negation ($\neg A$), and number restrictions ($\leqslant n\, r$ and $\geqslant n\, r$) can be handled.

An $\mathcal{FL}_0$ concept description is in *normal form* iff it is of the form

$$A_1 \sqcap \ldots \sqcap A_m \sqcap \forall r_1.C_1 \sqcap \ldots \sqcap \forall r_n.C_n,$$

where $A_1, \ldots, A_m$ are distinct concept names, $r_1, \ldots, r_n$ are *distinct* role names, and $C_1, \ldots, C_n$ are $\mathcal{FL}_0$ concept descriptions in normal form. It is easy to see that any description can be transformed into an equivalent one in normal form, using associativity,

commutativity and idempotence of $\sqcap$, and the fact that the descriptions $\forall r.(C \sqcap D)$ and $(\forall r.C) \sqcap (\forall r.D)$ are equivalent. Now, let

$$C \equiv A_1 \sqcap \ldots \sqcap A_m \sqcap \forall r_1.C_1 \sqcap \ldots \sqcap \forall r_n.C_n \text{ and } D \equiv B_1 \sqcap \ldots \sqcap B_k \sqcap \forall s_1.D_1 \sqcap \ldots \sqcap \forall s_\ell.D_\ell$$

respectively be the normal forms of the $\mathcal{FL}_0$ concept descriptions C and D. Then $C \sqsubseteq D$ iff the following two conditions hold:

1. for all $i, 1 \leq i \leq k$, there exists $j, 1 \leq j \leq m$ such that $B_i = A_j$.
2. For all $i, 1 \leq i \leq \ell$, there exists $j, 1 \leq j \leq n$ such that $s_i = r_j$ and $C_j \sqsubseteq D_i$.

It is easy to see that this characterization of subsumption is sound (i.e., the "if" direction of the equivalence holds) and complete (i.e., the "only-if" direction of the equivalence holds as well). This characterization yields an obvious recursive algorithm for computing subsumption. This algorithm can easily be shown to be of polynomial time complexity: in Condition 2, there is at most one subsumption test per role name occurring in C and D since all role names in $r_1, \ldots, r_n$ and all role names in $s_1, \ldots, s_\ell$ are distinct.

If we extend $\mathcal{FL}_0$ by language constructors that can express unsatisfiable concepts, then we must, on the one hand, change the definition of the normal form. On the other hand, the structural comparison of the normal forms must take into account that an unsatisfiable concept is subsumed by every concept. The simplest DL where this occurs is $\mathcal{FL}_\bot$, the extension of $\mathcal{FL}_0$ by the bottom concept $\bot$. An $\mathcal{FL}_\bot$ concept description is in *normal form* iff it is $\bot$ or of the form

$$A_1 \sqcap \ldots \sqcap A_m \sqcap \forall R_1.C_1 \sqcap \ldots \sqcap \forall R_n.C_n,$$

where $A_1, \ldots, A_m$ are distinct concept names different from $\bot$, $R_1, \ldots, R_n$ are distinct role names, and $C_1, \ldots, C_n$ are $\mathcal{FL}_\bot$ concept descriptions in normal form. Again, such a normal form can easily be computed. In principle, one just computes the $\mathcal{FL}_0$-normal form of the description (where $\bot$ is treated as an ordinary concept name): $B_1 \sqcap \ldots \sqcap B_k \sqcap \forall R_1.D_1 \sqcap \ldots \sqcap \forall R_n.D_n$. If one of the B_is is $\bot$, then replace the whole description by $\bot$. Otherwise, apply the same procedure recursively to the D_js. For example, the $\mathcal{FL}_0$-normal form of $\forall R.\forall R.B \sqcap A \sqcap \forall R.(A \sqcap \forall R.\bot)$ is $A \sqcap \forall R.(A \sqcap \forall R.(B \sqcap \bot))$, which yields the $\mathcal{FL}_\bot$-normal form $A \sqcap \forall R.(A \sqcap \forall R.\bot)$.

The structural subsumption algorithm for $\mathcal{FL}_\bot$ works just like the one for $\mathcal{FL}_0$, with the only difference that $\bot$ is subsumed by any description. For example, $\forall r.\forall r.B \sqcap A \sqcap \forall r.(A \sqcap \forall r.\bot) \sqsubseteq \forall r.\forall r.A \sqcap A \sqcap \forall r.A$ since the recursive comparison of their $\mathcal{FL}_\bot$-normal forms $A \sqcap \forall r.(A \sqcap \forall r.\bot)$ and $A \sqcap \forall r.(A \sqcap \forall r.A)$ finally leads to the comparison of $\bot$ and A.

The extension of $\mathcal{FL}_\bot$ by atomic negation (i.e., negation applied to concept names only) can be treated similarly. During the computation of the normal form, negated concept names are just treated like concept names. If, however, a name and its negation occur on the same level of the normal form, then $\bot$ is added, which can then be treated as described above. For example, $\forall r.\neg A \sqcap A \sqcap \forall r.(A \sqcap \forall r.B)$ is first transformed into $A \sqcap \forall r.(A \sqcap \neg A \sqcap \forall r.B)$, then into $A \sqcap \forall r.(\bot \sqcap A \sqcap \neg A \sqcap \forall r.B)$, and finally into $A \sqcap \forall r.\bot$. The structural comparison of the normal forms treats negated concept names just like concept names.

Finally, if we consider the language $\mathcal{ALN}$, the additional presence of number restrictions leads to a new type of conflict. On the one hand, as in the case of atomic negation,

number restrictions may be conflicting with each other (e.g., $\geqslant 2\, r$ and $\leqslant 1\, r$). On the other hand, at-least restrictions $\geqslant n\, r$ for $n \geq 1$ are in conflict with value restrictions $\forall r.\bot$ that prohibit role successors. When computing the normal form, one can again treat number restrictions like concept names, and then take care of the new types of conflicts by introducing $\bot$ and using it for normalization as described above. During the structural comparison of normal forms, one must also take into account inherent subsumption relationships between number restrictions (e.g., $\geqslant n\, r \sqsubseteq \geqslant m\, r$ iff $n \geq m$). A more detailed description of a structural subsumption algorithm working on a graph-like data structure for a DL extending $\mathcal{ALN}$ can be found in [40].

Satisfiability in other sub-Boolean DLs

Until now, we have only considered sub-Boolean DLs that extend $\mathcal{AL}$. Hemaspaandra [84] looks at all possible combinations of the constructors

$$\top,\ \ \bot,\ \ C \sqcap D,\ \ C \sqcup D,\ \ \neg A,\ \ \neg C,\ \ \forall r.C,\ \ \exists r.C,$$

and shows that there are only four possibilities for the complexity of the *satisfiability* problem:[12] P, NP-complete, coNP-complete, and PSPACE-complete:

- DLs that contain a complete basis for $\mathcal{ALC}$ have a PSPACE-complete satisfiability problem.
- DLs that contain a complete basis for propositional logic, but not for $\mathcal{ALC}$, have an NP-complete satisfiability problem.
- The DL $\mathcal{ALE}$ and its sublanguages where $\bot$, $\top$, or both are disallowed, have a coNP-complete satisfiability problem.
- The DL defined by the constructors $\bot$, $C \sqcap D$, $C \sqcup D$, $\forall r.C$, $\exists r.C$ has a PSPACE-complete satisfiability problem.
- All other DLs obtained as a combination of the above constructors have a polynomial satisfiability problem.

Subsumption in $\mathcal{EL}$

Recall that $\mathcal{EL}$ is defined by the constructors top concept ($\top$), conjunction ($C \sqcap D$), and existential restriction ($\exists r.C$). We show that subsumption of $\mathcal{EL}$ concept descriptions can be decided in polynomial time by reducing the subsumption problem to a combinatorial problem on trees. Any $\mathcal{EL}$ concept description C can be represented as a tree $\mathcal{G}_C$ whose edges are labeled with role names and whose nodes are labeled with sets of primitive concepts (where the empty set stands for $\top$). For example, the $\mathcal{EL}$ concept descriptions

$$\begin{aligned} C &:= P \sqcap \exists r.(\exists r.(P \sqcap Q) \sqcap \exists s.Q) \sqcap \exists r.(P \sqcap \exists s.P) \\ D &:= \exists r.(\exists r.P \sqcap \exists s.Q) \sqcap \exists r.P \end{aligned}$$

yield the $\mathcal{EL}$ description trees $\mathcal{G}_C$ and $\mathcal{G}_D$ depicted in Figure 7 (see [25] for a formal definition of the translation between $\mathcal{EL}$ concept descriptions and $\mathcal{EL}$ description trees).

Let C, D be $\mathcal{EL}$ concept descriptions. A *simulation* from $\mathcal{G}_D$ to $\mathcal{G}_C$ is a binary relation Z between the nodes of $\mathcal{G}_D$ and the nodes of $\mathcal{G}_C$ such that

[12] Unfortunately, the complexity of subsumption is not considered in [84].

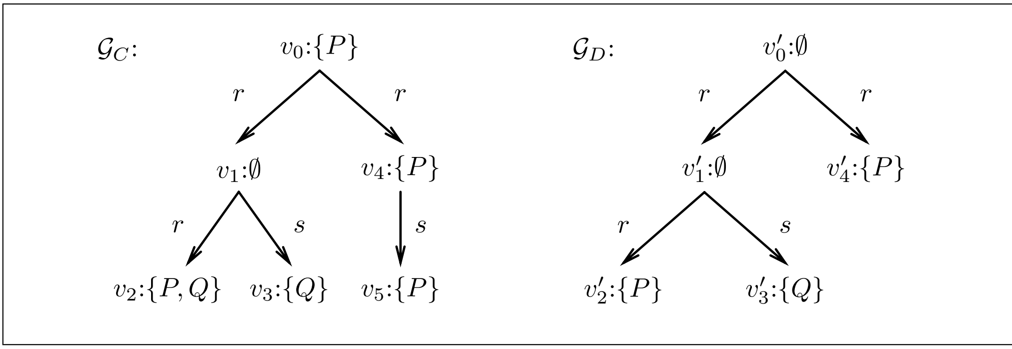


Figure 7. Two $\mathcal{EL}$ description trees.

1. $(v', v) \in Z$ implies that the label of v' is contained in the label of v;
2. $(v', v) \in Z$ implies that, for every r-successor u' of v' there is an r-successor u of v such that $(u', u) \in Z$.

Subsumption between $\mathcal{EL}$ concept descriptions corresponds to the existence of a simulation relation[13] between the corresponding trees: if v_0 is the root of $\mathcal{G}_C$ and v'_0 the root of $\mathcal{G}_D$, then we have

$$C \sqsubseteq D \quad \text{iff} \quad \text{there is a simulation } Z \text{ from } \mathcal{G}_D \text{ to } \mathcal{G}_C \text{ such that } (v'_0, v_0) \in Z.$$

In our example, we have $C \sqsubseteq D$ since the following relation Z is a simulation from $\mathcal{G}_D$ to $\mathcal{G}_C$:

$$Z := \{(v'_0, v_0),\ (v'_1, v_1),\ (v'_2, v_2),\ (v'_3, v_3),\ (v'_4, v_4)\}.$$

The definition of a simulation suggests the following top-down algorithm for constructing a simulation Z containing the tuple consisting of the roots (v'_0, v_0). First, put (v'_0, v_0) into Z and check whether the first property of a simulation (containment of labels) is satisfied for this tuple. If not, then stop with failure. Otherwise, try to extend Z by guessing pairs of successors of v'_0 and v_0, respectively, such that the second property of a simulation is satisfied for the pair (v'_0, v_0). Then continue the process with these new pairs. Since there are different ways of pairing off the successors, this algorithm is non-deterministic, and thus it does not yield a deterministic polynomial-time subsumption algorithm.

Fortunately, one can do better. One can compute the largest simulation between two trees (and actually also between two graphs with distinguished "root" nodes) by starting with all pairs of nodes, and then successively removing pairs that violate the first condition in the definition of a simulation or the second one (w.r.t. the current relation). It is easy to see that this procedure terminates after polynomially many steps with the largest simulation $\widehat{Z}$ from $\mathcal{G}_C$ to $\mathcal{G}_D$ (see [85] for a more efficient algorithm). We have $C \sqsubseteq D$ iff the pair consisting of the root nodes has not been removed, i.e., belongs to $\widehat{Z}$ (see [9] for more details).

[13] In [25], subsumption was actually characterized by the existence of a homomorphism, i.e., a simulation that is a total function. However, it is easy to see that, in case of *trees*, the existence of a simulation implies the existence of a homomorphism.

4.2 TBox reasoning in sub-Boolean DLs

As mentioned in Section 3, reasoning w.r.t. acyclic TBoxes can be reduced to reasoning on concept descriptions by expanding the definitions. Unfortunately, expansion may lead to an exponential blow-up of the descriptions. Is this due to the inherent complexity of reasoning with TBoxes, or can this exponential increase in the complexity be avoided? It turns out that the answer to this question depends on which sub-Boolean DL we consider.[14]

For the DL $\mathcal{FL}_0$, subsumption of concept descriptions is polynomial, whereas subsumption w.r.t. acyclic TBoxes is coNP-complete, subsumption w.r.t. cyclic TBoxes is PSPACE-complete, and subsumption w.r.t. GCIs in EXPTIME-complete. In contrast, for the DL $\mathcal{EL}$, subsumption remains polynomial even w.r.t. GCIs.

TBox reasoning in $\mathcal{FL}_0$

We start by describing an alternative approach for showing that subsumption of $\mathcal{FL}_0$ concept descriptions can be decided in polynomial time. In Section 4.1, the equivalence $\forall r.C \sqcap \forall r.D \equiv \forall r.(C \sqcap D)$ was used as a rewrite rule from left to right in order to compute the *structural subsumption normal form* of $\mathcal{FL}_0$ concept descriptions. If we use this rule in the opposite direction, we obtain a different normal form, which is called *concept-centered normal form* in [24], since it groups the concept descriptions w.r.t. concept names (and not w.r.t. role names, as the structural subsumption normal form does). Using this rule, any $\mathcal{FL}_0$ concept description can be transformed into an equivalent description that is a conjunction of descriptions of the form $\forall r_1. \cdots \forall r_m.A$ for $m \geq 0$ (not necessarily distinct) role names $r_1, \ldots, r_m$ and a concept name A. We abbreviate $\forall r_1. \cdots \forall r_m.A$ by $\forall r_1 \ldots r_m.A$, where $r_1 \ldots r_m$ is viewed as a word over the alphabet Σ of all role names. In addition, instead of $\forall w_1.A \sqcap \ldots \sqcap \forall w_\ell.A$ we write $\forall L.A$ where $L := \{w_1, \ldots, w_\ell\}$ is a finite set of words over Σ. The term $\forall \emptyset.A$ is considered to be equivalent to the top concept $\top$, which means that it can be added to a conjunction without changing the meaning of the concept. Using these abbreviations, any pair of $\mathcal{FL}_0$ concept descriptions C, D containing the concept names $A_1, \ldots, A_k$ can be rewritten as

$$C \equiv \forall U_1.A_1 \sqcap \ldots \sqcap \forall U_k.A_k \quad \text{and} \quad D \equiv \forall V_1.A_1 \sqcap \ldots \sqcap \forall V_k.A_k,$$

where U_i, V_i are finite sets of words over the alphabet of all role names. This normal form provides us with the following *characterization of subsumption* of $\mathcal{FL}_0$ concept descriptions [28]:

$$C \sqsubseteq D \quad \text{iff} \quad U_i \supseteq V_i \ \text{ for all } i, 1 \leq i \leq k.$$

Since the size of the concept-based normal forms is polynomial in the size of the original descriptions, and since the inclusion tests $U_i \supseteq V_i$ can also be realized in polynomial time, this yields a polynomial-time decision procedure for subsumption in $\mathcal{FL}_0$. In fact, as shown in [24], the structural subsumption algorithm for $\mathcal{FL}_0$ can be seen as a special implementation of these inclusion tests.

This characterization of subsumption via inclusion of finite sets of words can be extended to cyclic TBoxes with greatest fixpoint semantics as follows. A given TBox $\mathcal{T}$ can

[14] The same is true for propositionally closed DLs (see Section 6).

$$
\begin{array}{rcl}
A & \equiv & \forall r.A \sqcap \forall s.C \\
B & \equiv & \forall r.\forall s.C \\
C & \equiv & P \sqcap \forall s.C
\end{array}
$$

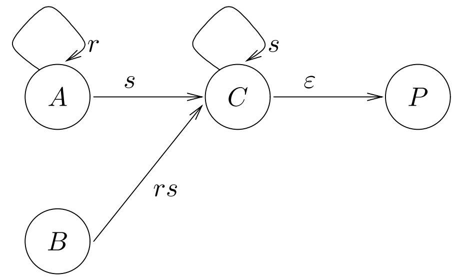


Figure 8. A cyclic $\mathcal{FL}_0$ TBox and the corresponding automaton.

be translated into a finite automaton[15] $\mathcal{A}_\mathcal{T}$ whose states are the concept names occurring in $\mathcal{T}$ and whose transitions are induced by the value restrictions occurring in $\mathcal{T}$ (see Fig. 8 for an example and [5] for the formal definition). For a defined concept A and a primitive concept P in $\mathcal{T}$, the language $L_{\mathcal{A}_\mathcal{T}}(A, P)$ is the set of all words labeling paths in $\mathcal{A}_\mathcal{T}$ from A to P. The languages $L_{\mathcal{A}_\mathcal{T}}(A, P)$ represent all the value restrictions that must be satisfied by instances of the concept A. With this intuition in mind, it should not be surprising that subsumption w.r.t. cyclic $\mathcal{FL}_0$ TBoxes can be characterized in terms of inclusion of regular languages represented by automata. Indeed, the following *characterizes subsumption w.r.t. greatest fixpoint semantics*:

$$A \sqsubseteq_{gfp,\mathcal{T}} B \quad \text{iff} \quad L_{\mathcal{A}_\mathcal{T}}(A, P) \supseteq L_{\mathcal{A}_\mathcal{T}}(B, P) \text{ for all primitive concepts } P.$$

In the example of Fig. 8, we have $L_{\mathcal{A}_\mathcal{T}}(A, P) = r^*ss^* \supset rss^* = L_{\mathcal{A}_\mathcal{T}}(B, P)$, and thus $A \sqsubseteq_\mathcal{T} B$, but not $B \sqsubseteq_\mathcal{T} A$.

Obviously, the languages $L_{\mathcal{A}_\mathcal{T}}(A, P)$ are regular, and any regular language can be obtained as such a language. Since inclusion of regular languages is a PSpace-complete problem [73], this shows that subsumption w.r.t. cyclic $\mathcal{FL}_0$ TBoxes with greatest fixpoint semantics is PSpace-complete [5]. The same complexity can be shown for subsumption in cyclic $\mathcal{FL}_0$ TBoxes interpreted with least fixpoint semantics or with descriptive semantics [5, 102]. In addition, the PSpace-completeness result can be extended to the DL $\mathcal{ALN}$ [106].

For an acyclic $\mathcal{FL}_0$ TBox $\mathcal{T}$, the automaton $\mathcal{A}_\mathcal{T}$ is acyclic as well. Since inclusion of languages accepted by acyclic finite automata is coNP-complete [73] and subsumption w.r.t. greatest fixpoint semantics coincides with subsumption w.r.t. descriptive semantics in the case of acyclic TBoxes, this proves Nebel's result that subsumption w.r.t. acyclic $\mathcal{FL}_0$-TBoxes is coNP-complete [128]. Thus, for $\mathcal{FL}_0$, even the presence of acyclic TBoxes increases the complexity of the subsumption problem.

Finally, ExpTime-hardness of subsumption in $\mathcal{FL}_0$ w.r.t. GCIs was shown in [10]. The ExpTime-upper bound follows from the fact that subsumption in $\mathcal{ALC}$ w.r.t. GCIs is in ExpTime [138].

[15] Strictly speaking, we obtain a finite automaton with word transitions, i.e., transitions that may be labeled by a word over Σ rather than a letter of Σ.

$$B \equiv P \sqcap \exists r.C,$$
$$C \equiv Q \sqcap \exists r.D, \qquad D \equiv P \sqcap \exists r.C,$$
$$A \equiv P \sqcap Q \sqcap \exists r.A', \quad A' \equiv Q \sqcap P \sqcap \exists r.D.$$

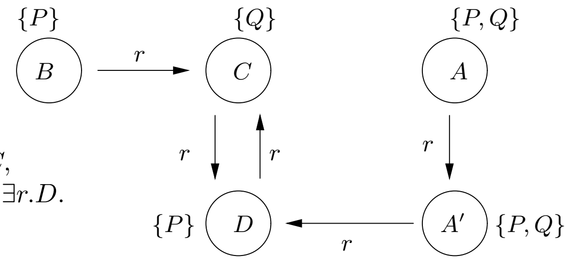


Figure 9. A normalized $\mathcal{EL}$ TBox and the corresponding description graph.

TBox reasoning in $\mathcal{EL}$

The approach for deciding subsumption between $\mathcal{EL}$ concept descriptions sketched in Section 4.1 can be extended to $\mathcal{EL}$ TBoxes [9]. In fact, one can show that any $\mathcal{EL}$ TBox can be transformed in polynomial time into an equivalent *normalized TBox* whose definitions are of the form

$$A \equiv P_1 \sqcap \ldots \sqcap P_m \sqcap \exists r_1.B_1 \sqcap \ldots \sqcap \exists r_\ell.B_\ell,$$

where $P_1, \ldots, P_m$ are primitive concepts, $r_1, \ldots, r_\ell$ roles, and $B_1, \ldots, B_\ell$ defined concepts. Any normalized $\mathcal{EL}$ TBox $\mathcal{T}$ can then be transformed into an $\mathcal{EL}$ *description graph* $\mathcal{G}_\mathcal{T}$ whose nodes are the defined concepts of $\mathcal{T}$. If A is a defined concept whose definition in $\mathcal{T}$ is of the normalized form shown above, then A has label $\{P_1, \ldots, P_m\}$, and is the source of the edges $(A, r_1, B_1), \ldots, (A, r_\ell, B_\ell)$ (see Figure 9 for an example).

Subsumption w.r.t. greatest fixpoint semantics corresponds to the existence of an appropriate simulation on $\mathcal{G}_\mathcal{T}$ [9]: if A, B are defined concepts in $\mathcal{T}$, then

$$A \sqsubseteq_{gfp,\mathcal{T}} B \quad \text{iff} \quad \text{there is a simulation } Z \text{ from } \mathcal{G}_\mathcal{T} \text{ to } \mathcal{G}_\mathcal{T} \text{ such } (B, A) \in Z.$$

Since the algorithm for computing the largest simulation sketched in Section 4.1 also works on graphs, this shows that subsumption w.r.t. $\mathcal{EL}$ TBoxes interpreted with greatest fixpoint semantics can be decided in polynomial time. In [9], the same result is also shown for descriptive and least fixpoint semantics. As a special case we have that subsumption w.r.t. acyclic $\mathcal{EL}$ TBoxes is polynomial.

In [46], it is shown that subsumption in $\mathcal{EL}$ remains polynomial even in the presence of GCIs, and in [10] this result is extended to the DL $\mathcal{EL}^{++}$, which extends $\mathcal{EL}$ by the bottom concept, nominals, a restricted form of concrete domains, and a restricted form of role-value maps.[16]

The polynomial-time subsumption algorithms for $\mathcal{EL}$ and $\mathcal{EL}^{++}$ actually classify the given set of GCIs $\mathcal{T}$, i.e., they simultaneously compute all subsumption relationships between the concept names occurring in $\mathcal{T}$. In the following, we sketch an algorithm for $\mathcal{EL}$. This algorithm proceeds in four steps:

1. Normalize the set of GCIs.

2. Translate the normalized set of GCIs into a graph.

[16] Concrete domains and role-value maps will be introduced in Section 6. Adding unrestricted concrete domains or role-value maps to $\mathcal{EL}$ with GCIs would cause undecidability of subsumption [10, 8].

3. Complete the graph using completion rules.

4. Read off the subsumption relationships from the normalized graph.

A set of $\mathcal{EL}$ GCIs is *normalized* iff it only contains GCIs of the following form:

$$A_1 \sqcap A_2 \sqsubseteq B, \quad A \sqsubseteq \exists r.B, \quad \exists r.A \sqsubseteq B,$$

where A, A_1, A_2, B are concept names or the top-concept $\top$. One can transform a given set of GCIs into a normalized one by applying normalization rules. Instead of describing these rules in the general case, we just illustrate them by an example:

$$\begin{array}{rl}
\exists r.A \sqcap \exists r.\exists s.A \sqsubseteq A \sqcap \exists r.\top & \rightsquigarrow \quad \exists r.A \sqsubseteq B_1, \quad B_1 \sqcap \exists r.\exists s.A \sqsubseteq A \sqcap \exists r.\top \\
\rightsquigarrow & \exists r.A \sqsubseteq B_1, \quad \exists r.\exists s.A \sqsubseteq B_2, \quad B_1 \sqcap B_2 \sqsubseteq A \sqcap \exists r.\top \\
\rightsquigarrow & \exists r.A \sqsubseteq B_1, \quad \exists s.A \sqsubseteq B_3, \quad \exists r.B_3 \sqsubseteq B_2, \quad B_1 \sqcap B_2 \sqsubseteq A \sqcap \exists r.\top \\
\rightsquigarrow & \exists r.A \sqsubseteq B_1, \quad \exists s.A \sqsubseteq B_3, \quad \exists r.B_3 \sqsubseteq B_2, \quad B_1 \sqcap B_2 \sqsubseteq A, \quad B_1 \sqcap B_2 \sqsubseteq \exists r.\top
\end{array}$$

For example, in the first normalization step we introduce the abbreviation B_1 for the description $\exists r.A$. One might think that one must make B_1 equivalent to $\exists r.A$, i.e., also add the GCI $B_1 \sqsubseteq \exists r.A$. However, it can be shown that adding just $\exists r.A \sqsubseteq B_1$ is sufficient to obtain a *subsumption-equivalent* set of GCIs, i.e., a set that induces the same subsumption relationships between the concept names occurring in the original set of GCIs. All normalization rules preserve equivalence in this sense, and if one uses an appropriate strategy (which basically defers the applications of the rule applied in the last step of our example to the end), then the normal form can be computed in linear time.

In the next step, we build the *classification graph* $G_\mathcal{T} = (V, V \times V, S, R)$ where

- V is the set of concept names (including $\top$) occurring in the normalized set of GCIs $\mathcal{T}$;
- S labels nodes with sets of concept names (again including $\top$);
- R labels edges with sets of role names.

The label sets are supposed to satisfy the following *invariants*:

- $B \in S(A)$ implies $A \sqsubseteq_\mathcal{T} B$, i.e., $S(A)$ contains only subsumers of A w.r.t. the set of GCIs $\mathcal{T}$.
- $r \in R(A, B)$ implies $A \sqsubseteq_\mathcal{T} \exists r.B$, i.e., $R(A, B)$ contains only roles r such that $\exists r.B$ subsumes A.

Initially, we set $S(A) := \{A, \top\}$ for all nodes $A \in V$, and $R(A, B) := \emptyset$ for all edges $(A, B) \in V \times V$. Obviously, the above invariants are satisfied by these initial label sets.

The labels of nodes and edges are then extended by applying the rules of Figure 10. Note that such a rule is only applied if it really extends a label set. It is easy to see that these rules preserve the above invariants. For example, consider the (most complicated) rule (R3). Obviously, $\exists r.B_1 \sqsubseteq A_1 \in \mathcal{T}$ implies $\exists r.B_1 \sqsubseteq_\mathcal{T} A_1$, and the assumption that the invariants are satisfied before applying the rule yields $B \sqsubseteq_\mathcal{T} B_1$ and $A \sqsubseteq_\mathcal{T} \exists r.B$. The subsumption relationship $B \sqsubseteq_\mathcal{T} B_1$ obviously implies $\exists r.B \sqsubseteq_\mathcal{T} \exists r.B_1$. By applying transitivity of the subsumption relation $\sqsubseteq_\mathcal{T}$, we thus obtain $A \sqsubseteq_\mathcal{T} A_1$.

The fact that subsumption in $\mathcal{EL}$ w.r.t. GCIs can be decided in polynomial time is an immediate consequence of the following statements:

(R1)	$A_1 \sqcap A_2 \sqsubseteq B \in \mathcal{T}$ and $A_1, A_2 \in S(A)$		then	add B to $S(A)$
(R2)	$A_1 \sqsubseteq \exists r.B \in \mathcal{T}$ and $A_1 \in S(A)$		then	add r to $R(A, B)$
(R3)	$\exists r.B_1 \sqsubseteq A_1 \in \mathcal{T}$ and $B_1 \in S(B), r \in S(A, B)$		then	add A_1 to $S(A)$

Figure 10. The completion rules for subsumption in $\mathcal{EL}$ w.r.t. GCIs.

1. Rule application terminates after a polynomial number of steps.

2. If no more rules are applicable, then $A \sqsubseteq_{\mathcal{T}} B$ iff $B \in S(A)$.

Regarding the first statement, note that the number of nodes is linear and the number of edges is quadratic in the size of $\mathcal{T}$. In addition, the size of the label sets is bounded by the number of concept names and role names, and each rule application extends at least one label. Regarding the equivalence in the second statement, the "if" direction follows from the fact that the above invariants are preserved under rule application. To show the "only-if" direction, assume that $B \notin S(A)$. Then the following interpretation $\mathcal{I}$ is a model of $\mathcal{T}$ in which $A \in A^{\mathcal{I}}$, but $A \notin B^{\mathcal{I}}$:

- $\Delta^{\mathcal{I}} := V$;
- $r^{\mathcal{I}} := \{(A', B') \mid r \in R(A', B')\}$ for all role names r;
- $B'^{\mathcal{I}} := \{A' \mid B' \in S(A')\}$ for all concept names A'.

More details can be found in [46, 10].

4.3 ABox reasoning in sub-Boolean DLs

In [67], the complexity of instance checking in DLs of the $\mathcal{AL}$ family is investigated. With one exception, the complexity[17] of instance checking coincides with the complexity of subsumption. This one exception is $\mathcal{ALE}$, where the subsumption problem is NP-complete, whereas instance checking is PSPACE-complete. In the following, we sketch the PSPACE-hardness proof given in [67]. It depends on the PSPACE-hardness proof of satisfiability in $\mathcal{ALC}$ given in [143], which works by a reduction[18] from Quantified Boolean Formulae (QBF), whose validity problem is known to be PSPACE-complete [73]. A given QBF φ is translated in polynomial time into an $\mathcal{ALC}$ concept description C_φ such that φ is valid iff C_φ is satisfiable. For the purpose of sketching the PSPACE-hardness proof from [67], it is not really necessary to know what a QBF is and how the reduction concept C_φ is defined in detail. The first important observation made in [67] is that C_φ is equivalent to a concept description of the form

$$D \sqcap \neg D_1 \sqcap \ldots \sqcap \neg D_n$$

[17] To be more precise, the "combined complexity" of $\mathcal{A} \models C(a)$, i.e., w.r.t. both the size of $\mathcal{A}$ and the size of C.

[18] Note that this reduction actually differs from the one usually employed in modal logic to show PSPACE-hardness of K [81].

where $D, D_1, \ldots, D_n$ are $\mathcal{ALE}$ concept descriptions whose size is polynomially related to the size of C_φ. Thus, it remains to be shown that satisfiability of concept descriptions of this form can be reduced in polynomial time to instance checking in $\mathcal{ALE}$.

Assume that $\mathcal{ALE}$ concept descriptions $D, D_1, \ldots, D_n$ are given. Let q be a new role and E_n a new concept name. We define $\mathcal{ALE}$ concept descriptions $E_0, \ldots, E_{n-1}$ as follows:

$$E_i := \exists q.(D_{i+1} \sqcap E_{i+1}) \quad \text{for } i = 0, \ldots, n-1.$$

The ABox $\mathcal{A}$ is defined as

$$\mathcal{A} \;:=\; \{\, q(a,a),\ q(a,b),\quad D_1(a),\ \ldots,\ D_n(a),\quad E_1(b),\ \ldots,\ E_n(b),\quad D(b)\,\}$$

In [67], it is shown that

$$D \sqcap \neg D_1 \sqcap \ldots \sqcap \neg D_n \text{ is satisfiable} \quad \text{iff} \quad \mathcal{A} \not\models E_0(a).$$

In fact, assume that $\mathcal{A} \not\models E_0(a)$. Then there is a model $\mathcal{I}$ of $\mathcal{A}$ such that

$$a^{\mathcal{I}} \in (\neg E_0)^{\mathcal{I}} = (\forall q.(\neg D_1 \sqcup \neg E_1))^{\mathcal{I}}.$$

Thus $(a^{\mathcal{I}}, a^{\mathcal{I}}) \in q^{\mathcal{I}}, a^{\mathcal{I}} \in D_1^{\mathcal{I}}$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in q^{\mathcal{I}}, b^{\mathcal{I}} \in E_1$ imply that $a^{\mathcal{I}} \in (\neg E_1)^{\mathcal{I}}$ and $b^{\mathcal{I}} \in (\neg D_1)^{\mathcal{I}}$. We can now apply the same argument to $a^{\mathcal{I}} \in (\neg E_1)^{\mathcal{I}} = (\forall q.(\neg D_2 \sqcup \neg E_2))^{\mathcal{I}}$, etc. In the end, we obtain that $b^{\mathcal{I}} \in (\neg D_i)^{\mathcal{I}}$ for $i = 1, \ldots, n$, and since we also have $b^{\mathcal{I}} \in D^{\mathcal{I}}$ this shows that $D \sqcap \neg D_1 \sqcap \ldots \sqcap \neg D_n$ is satisfiable.

Conversely, it is easy to see that a model of $D \sqcap \neg D_1 \sqcap \ldots \sqcap \neg D_n$ can be used to construct a model of $\mathcal{A}$ in which a does not belong to E_0.

4.4 Bi-simulation characterizations of sub-Boolean DLs

As noted before, concept descriptions of the $\mathcal{AL}$ family (and also of many other DLs) can be translated into first-order formulae with one free variable. Thus, any such DL $\mathcal{L}$ yields a fragment $FO_{\mathcal{L}}$ of first-order predicate logic, which consists of those formulae with one free variable that are equivalent to the first-order translation of an $\mathcal{L}$ concept description. These fragments can be used to compare the *expressive power*[19] of DLs.

We say that a DL $\mathcal{L}_2$ is *strictly more expressive* than a DL $\mathcal{L}_1$ ($\mathcal{L}_1 \prec \mathcal{L}_2$) iff $FO_{\mathcal{L}_1} \subset FO_{\mathcal{L}_2}$, i.e., every first-order translation of an $\mathcal{L}_1$ concept description is equivalent to the first-order translation of an $\mathcal{L}_2$ concept description, and there is an $\mathcal{L}_2$ concept description whose translation is not equivalent to any translation of an $\mathcal{L}_1$ concept description.

Usually, the inclusion between two fragments $FO_{\mathcal{L}_1}$ and $FO_{\mathcal{L}_2}$ is relatively easy to show. However, how can one show that such an inclusion is strict? One way of doing this is to use an appropriate bisimulation characterization of the first-order fragments. For example, it is well-know that the fragment $FO_{\mathcal{ALC}}$ consists of those first-order formulae that are preserved under bisimulation (see Chapter 1). This can be used to show that $\mathcal{ALC} \prec \mathcal{ALCN}$ by giving an example of an $\mathcal{ALCN}$ concept description that is not preserved under bisimulation (see [105]).

In [105], the bisimulation characterization of the first-order fragment corresponding to $\mathcal{ALC}$ is adapted to various sub-Boolean DLs, and then used to compare their expressive power. Here, we only sketch the characterization of $FO_{\mathcal{AL}}$.

[19]The definition of the expressive power of DLs obtained this way is used in [105]; it is weaker than the one defined in [4] since it does not allow one to extend the vocabulary.

First, we must introduce some notation. If X, Y are subsets of a set Δ, and R is a binary relation on Δ, then we define

$$\begin{array}{lll} X\,R^{\uparrow}\,Y & \text{iff} & \text{for all } d \in X \text{ there is } e \in Y \text{ such that } (d,e) \in R, \\ X\,R^{\downarrow}\,Y & \text{iff} & \text{for all } e \in Y \text{ there is } d \in X \text{ such that } (d,e) \in R. \end{array}$$

In order to explain the intuition underlying this definition from a DL point of view, assume that r is a role, C, D are concept descriptions, and $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is an interpretation, and define $R := r^{\mathcal{I}}$, $X := C^{\mathcal{I}}$, and $Y := D^{\mathcal{I}}$. Then $X\,R^{\uparrow}\,Y$ means that $C^{\mathcal{I}} \subseteq (\exists r.D)^{\mathcal{I}}$, and $X\,R^{\downarrow}\,Y$ means that $D^{\mathcal{I}} \subseteq (\exists r^{-}.C)^{\mathcal{I}}$.

Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ be two interpretations. An *$\mathcal{AL}$-simulation* between $\mathcal{I}$ and $\mathcal{J}$ is a non-empty relation $Z \subseteq 2^{\Delta^{\mathcal{I}}} \times \Delta^{\mathcal{J}}$ such that the following three conditions are satisfied:

1. If $(X_1, d_2) \in Z$, then $X_1 \subseteq A^{\mathcal{I}}$ implies $d_2 \in A^{\mathcal{J}}$ and $X_1 \subseteq (\neg A)^{\mathcal{I}}$ implies $d_2 \in (\neg A)^{\mathcal{J}}$ for all concept names A.

2. For every role name r, if $(X_1, d_2) \in Z$ and $X_1\,(r^{\mathcal{I}})^{\uparrow}\,Y_1$, then there is an $e_2 \in \Delta^{\mathcal{J}}$ such that $(d_2, e_2) \in r^{\mathcal{I}}$.

3. For every role name r, if $(X_1, d_2) \in Z$ and $(d_2, e_2) \in r^{\mathcal{J}}$, then there is an $Y_1 \subseteq \Delta^{\mathcal{I}}$ such that $X_1\,(r^{\mathcal{I}})^{\downarrow}\,Y_1$ and $(Y_1, e_2) \in Z$.

Intuitively, the fact that $\mathcal{AL}$-simulations are binary relations between *subsets* of $\Delta^{\mathcal{I}}$ and $\Delta^{\mathcal{J}}$ makes sure that disjunction (which is not available in $\mathcal{AL}$) is not preserved. The first clause of the definition ensures that concept names and negated concept names (but not full negation) are preserved. The second clause ensures preservation of restricted existential restrictions $\exists r.\top$, but not of full existential restrictions $\exists r.C$ (since we do not require $(Y_1, e_2) \in Z$). The third clause ensures that value restrictions are preserved.

The first order formula $\alpha(x)$ is *preserved under $\mathcal{AL}$-simulations* iff for all interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ and all $\mathcal{AL}$-simulations Z between $\mathcal{I}$ and $\mathcal{J}$, we have:

$$(X, d_2) \in Z \text{ and } \mathcal{I} \models \alpha(d_1) \text{ for all } d_1 \in X \quad \text{implies} \quad \mathcal{I} \models \alpha(d_2).$$

In [105], it is shown that $FO_{\mathcal{AL}}$ consists of those first-order formulae that are preserved under $\mathcal{AL}$-simulations.

This result can be used to show that $\mathcal{ALU}$ is strictly more expressive than $\mathcal{AL}$. In fact, the formula $A(x) \vee B(x)$ obtained by translating the $\mathcal{ALU}$ concept description $A \sqcup B$ into first-order logic is not preserved under $\mathcal{AL}$-simulations. To see this, let $\mathcal{I}$ be the interpretation consisting of two elements d_1, e_1, where d_1 belongs to A and e_1 belongs to B, and let $\mathcal{J}$ be the interpretation consisting of the element d_2, which belongs neither to A nor to B. It is easy to see that $Z := \{(\{d_1, e_1\}, d_2)\}$ is an $\mathcal{AL}$-simulation between $\mathcal{I}$ and $\mathcal{J}$. However, $(\{d_1, e_1\}, d_2) \in Z$ and both d_1 and e_1 satisfy $A(x) \vee B(x)$, but d_2 does *not* satisfy $A(x) \vee B(x)$.

5 NON-STANDARD INFERENCES

After motivating the need for non-standard inferences in DLs and illustrating some of them by examples, we give formal definitions of the most important non-standard inferences considered until now, and review the existing results.

5.1 Motivation

All DL systems provide their users with standard inference services like computing the subsumption hierarchy, testing ABox consistency, and instance checking. These inferences are not only useful when working with “finished” knowledge bases, they can also support the knowledge engineer while building a knowledge base, by pointing out inconsistencies and unwanted consequences. They can help the knowledge engineer to check whether a concept definition makes sense, but they provide no support for actually coming up with a first version of the definition. The non-standard inferences introduced in this section can be used to overcome this deficit, basically by providing two ways of re-using “old” knowledge when defining new one: (i) constructing concepts by generalizing from examples, and (ii) constructing concepts by modifying “similar” ones.

The first approach was introduced as *bottom-up construction* of description logic knowledge bases in [20, 25]. Instead of defining the relevant concepts of an application domain from scratch, this methodology allows the user to give typical examples of individuals belonging to the concept to be defined. These individuals are then generalized to a concept by first computing the most specific concept (msc) of each individual (i.e., the least concept description w.r.t. subsumption in the available description language that has this individual as an instance), and then computing the least common subsumer (lcs) of these concepts (i.e., the least concept description w.r.t. subsumption in the available description language that subsumes all these concepts). The knowledge engineer can then use the computed concept as a starting point for the concept definition. As a simple example, assume that the knowledge engineer has already defined the concept of a man and a woman as

$$\mathsf{Man} \equiv \mathsf{Human} \sqcap \mathsf{Male} \quad \text{and} \quad \mathsf{Woman} \equiv \mathsf{Human} \sqcap \mathsf{Female},$$

and now wants to define the concept of a parent, but does not know how to do this within the available DL (which we assume to be $\mathcal{EL}$ in this example). However, the available ABox

$$\begin{array}{lll} \mathsf{Man}(\mathsf{JACK}), & \mathsf{child}(\mathsf{JACK}, \mathsf{CAROLINE}), & \mathsf{Woman}(\mathsf{CAROLINE}), \\ \mathsf{Woman}(\mathsf{JACKIE}), & \mathsf{child}(\mathsf{JACKIE}, \mathsf{JOHN}), & \mathsf{Man}(\mathsf{JOHN}), \end{array}$$

contains the individuals JACK and JACKIE, of whom the knowledge engineer knows that they are parents. The most specific concepts of JACK and JACKIE in the given ABox are

$$\mathsf{Man} \sqcap \exists \mathsf{child}.\mathsf{Woman} \quad \text{and} \quad \mathsf{Woman} \sqcap \exists \mathsf{child}.\mathsf{Man},$$

respectively, and the least common subsumer (in $\mathcal{EL}$) of these two concepts w.r.t. the definitions of Man and Woman is

$$\mathsf{Human} \sqcap \exists \mathsf{child}.\mathsf{Human},$$

which looks like a good starting point for a definition of parent.

In contrast to standard inferences like subsumption and instance checking, the output of the non-standard inferences we have mentioned until now (computing the msc and the lcs) is a concept description rather than a yes/no answer. In such a setting, it is important that the returned descriptions are as readable and comprehensible as possible. Unfortunately, the descriptions that are produced by the known algorithms for computing

the lcs and the msc do not satisfy this requirement. The reason is that – like most algorithms for the standard inference problems – these algorithms work on expanded concept descriptions, i.e., concept descriptions that do not contain names defined in the underlying TBox. Consequently, the descriptions that the algorithms produce also do not use defined concepts, which makes them in many cases large and hard to read and comprehend.[20] This problem can be overcome by *rewriting* the resulting concept w.r.t. the given TBox. Informally, the problem of rewriting a concept given a terminology can be stated as follows: given an acyclic TBox $\mathcal{T}$ and a concept description C that does not contain concept names defined in $\mathcal{T}$, can this description be rewritten into an equivalent shorter description E by using (some of) the names defined in $\mathcal{T}$? For example, w.r.t. the TBox in Figure 3, the concept description

$$\mathsf{Person} \sqcap \forall \mathsf{child}.\mathsf{Female} \sqcap \exists \mathsf{child}.\top \sqcap \forall \mathsf{child}.\mathsf{Person}$$

can be rewritten to the equivalent concept $\mathsf{Parent} \sqcap \forall \mathsf{child}.\mathsf{Woman}$.

Rewriting w.r.t. a TBox is just one instance of a more general rewriting framework, which will be introduced below. Another instance of this framework is *approximation*, where one tries to express a concept description C_1 defined in one DL $\mathcal{L}_1$ by a concept description C_2 expressed in another DL $\mathcal{L}_2$. If $\mathcal{L}_1$ is strictly more expressive than $\mathcal{L}_2$, then it is not always possible to find a concept description C_2 that is equivalent to C_1. In this case, one can try to approximate C_1 by an $\mathcal{L}_2$ concept description that is as "close" as possible to C_1, for example by trying to find an $\mathcal{L}_2$ concept description that subsumes C_1 and is minimal w.r.t. subsumption. One possible application for such an inference is translating knowledge bases from the language employed by one system into the language employed by another system.

In order to apply the second approach of constructing concepts by modifying existing ones, one must first find the right candidates for modification. One way of doing this is to give a partial description of the concept to be defined as a concept pattern (i.e., a concept description containing variables standing for concept descriptions), and then look for concept descriptions that match this pattern. For example, the pattern

$$\mathsf{Man} \sqcap \exists \mathsf{child}.(\mathsf{Man} \sqcap X) \sqcap \exists \mathsf{spouse}.(\mathsf{Woman} \sqcap X)$$

looks for descriptions of classes of men whose wives and sons share some characteristic. An example of a concept description *matching* this pattern is $\mathsf{Man} \sqcap \exists \mathsf{child}.(\mathsf{Man} \sqcap \mathsf{Tall}) \sqcap \exists \mathsf{spouse}.(\mathsf{Woman} \sqcap \mathsf{Tall})$.

Unification is a generalization of matching where both concepts may contain variables. The main motivation for introducing unification in DLs was to avoid redundancies in knowledge bases that are built by several knowledge engineers over a long time period. In this setting, it frequently happens that the same (intuitive) concept is introduced several times, often with slightly differing descriptions. Testing for equivalence of concepts is not always sufficient to find out whether, for a given concept description, there already exists another concept description in the knowledge base describing the same notion. As an example, lets us ask whether the following two $\mathcal{FL}_0$ concept descriptions might denote

[20] In the above example, this means that the definitions of Man and Woman are expanded before applying the lcs algorithm. If Human also had a definition, then it would also be expanded, and instead of the concept description containing Human shown above, the algorithm would return its expanded version.

the same (intuitive) concept:

$$\forall \mathsf{child}.\forall \mathsf{child}.\mathsf{Rich} \sqcap \forall \mathsf{child}.\mathsf{Rmr} \quad \text{and} \quad \mathsf{Acr} \sqcap \forall \mathsf{child}.\mathsf{Acr} \sqcap \forall \mathsf{child}.\forall \mathsf{spouse}.\mathsf{Rich}.$$

The answer is yes, since replacing the concept name Rmr by the description $\mathsf{Rich} \sqcap \forall \mathsf{spouse}.\mathsf{Rich}$ and Acr by $\forall \mathsf{child}.\mathsf{Rich}$ yields the descriptions

$$\begin{array}{l}\forall \mathsf{child}.\forall \mathsf{child}.\mathsf{Rich} \sqcap \forall \mathsf{child}.(\mathsf{Rich} \sqcap \forall \mathsf{spouse}.\mathsf{Rich}),\\ \forall \mathsf{child}.\mathsf{Rich} \sqcap \forall \mathsf{child}.\forall \mathsf{child}.\mathsf{Rich} \sqcap \forall \mathsf{child}.\forall \mathsf{spouse}.\mathsf{Rich},\end{array}$$

which are obviously equivalent. Thus, under the assumption that Rmr stands for "Rich and married rich" and Acr for "All children are rich", we can conclude that both descriptions are meant to express the concept "All grandchildren are rich and all children are rich and married rich". This connection between the two description can be found by a unification algorithm if we declare Rmr and Acr to be variables. Of course, unifiability does not necessarily mean that the concept descriptions are meant to represent the same concept. Unifiability only suggests that there is a possible connection: the final decision must be taken by the knowledge engineer.

5.2 Least common subsumers and most specific concepts

Intuitively, the least common subsumer of a given collection of concept descriptions is a description that represents the properties that all the elements of the collection have in common. More formally, it is the most specific concept description that subsumes the given descriptions. What this most specific description looks like, whether it really captures the intuition of representing the properties common to the input descriptions, and whether it exists at all strongly depends on the DL under consideration.

Let $\mathcal{L}$ be a DL. A concept description E of $\mathcal{L}$ is a *least common subsumer* (lcs) of the concept descriptions $C_1, \ldots, C_n$ in $\mathcal{L}$ ($lcs_{\mathcal{L}}(C_1, \ldots, C_n)$ for short) iff it satisfies

1. $C_i \sqsubseteq E$ for all $i = 1, \ldots, n$, and
2. E is the least $\mathcal{L}$ concept description with this property, i.e., if E' is an $\mathcal{L}$ concept description satisfying $C_i \sqsubseteq E'$ for all $i = 1, \ldots, n$, then $E \sqsubseteq E'$.

As an easy consequence of this definition, the lcs is unique up to equivalence, which justifies talking about *the* lcs. In addition, the n-ary lcs as defined above can be reduced to the binary lcs (the case where $n = 2$). Indeed, it is easy to see that $lcs_{\mathcal{L}}(C_1, \ldots, C_n) \equiv lcs_{\mathcal{L}}(C_1, \ldots, lcs_{\mathcal{L}}(C_{n-1}, C_n) \cdots)$. Thus, it is enough to devise algorithms for computing the binary lcs.

It should be noted, however, that the lcs need not always exist. This can have different reasons: (a) there may not exist a concept description in $\mathcal{L}$ satisfying (i) of the definition (i.e., subsuming $C_1, \ldots, C_n$); (b) there may be several subsumption incomparable minimal concept descriptions satisfying (i) of the definition; (c) there may be an infinite chain of more and more specific descriptions satisfying (i) of the definition. Obviously, (a) cannot occur for DLs containing the top concept. It is easy to see that, for DLs allowing for conjunction of descriptions, (b) cannot occur. An example for a DL exhibiting behavior (c) can be found in [6], where the lcs is defined w.r.t. a cyclic TBox.

It is also clear that in DLs allowing for disjunction, the lcs of $C_1, \ldots, C_n$ is their disjunction $C_1 \sqcup \ldots \sqcup C_n$. In this case, the lcs is not really of interest. Instead of

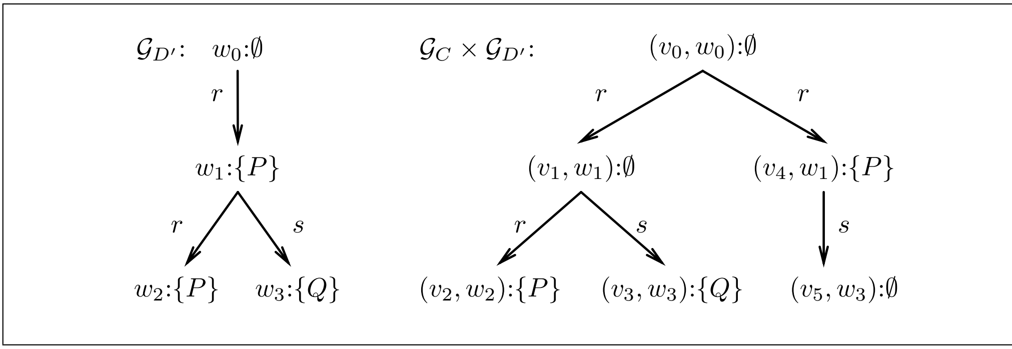


Figure 11. The product of $\mathcal{EL}$ description trees.

extracting properties common to $C_1, \ldots, C_n$, it just gives their disjunction, which does not provide us with new information. Thus, it only makes sense to look at the lcs in sub-Boolean DLs.

For DLs whose expressive power lies between $\mathcal{FL}_0$ and $\mathcal{ALN}$, one can use the characterization of subsumption via finite languages over the alphabet of the role names (see Subsection 4.2) to compute the lcs. Recall that any pair of $\mathcal{FL}_0$ concept descriptions C, D containing the concept names $A_1, \ldots, A_k$ can be written as

$$C \equiv \forall U_1.A_1 \sqcap \ldots \sqcap \forall U_k.A_k \quad \text{and} \quad D \equiv \forall V_1.A_1 \sqcap \ldots \sqcap \forall V_k.A_k,$$

where U_i, V_i are finite sets of words over the alphabet of all role names, and that $C \sqsubseteq D$ iff $U_i \supseteq V_i$ for $i = 1, \ldots, k$. As an easy consequence of this characterization we obtain that the lcs E of C, D is of the form

$$E \equiv \forall (U_1 \cap V_1).A_1 \sqcap \ldots \sqcap \forall (U_k \cap V_k).A_k.$$

Using the language-based characterization of subsumption in $\mathcal{ALN}$ [106], this approach for computing the lcs by language intersection can be extended to $\mathcal{ALN}$ [20], but this involves the use of certain infinite regular languages.

For DLs with existential restrictions, the characterization of subsumption via the existence of certain simulation relations between description trees (see Subsection 4.2) implies that the lcs corresponds to the product of the description trees [25]. The *product* $\mathcal{G}_C \times \mathcal{G}_D$ of two $\mathcal{EL}$ description trees $\mathcal{G}_C$ and $\mathcal{G}_D$ is defined by induction on the depth of the trees. Its root is the pair (v_0, w_0) consisting of the roots of $\mathcal{G}_C$ and $\mathcal{G}_D$, and the label of (v_0, w_0) is the intersection of the labels of v_0 and w_0. For each r-successor v of v_0 in $\mathcal{G}_C$ and w of w_0 in $\mathcal{G}_D$, we obtain an r-successor (v, w) of (v_0, w_0) in $\mathcal{G}_C \times \mathcal{G}_D$ that is the root of the product of the subtree of $\mathcal{G}_C$ with root v and the subtree of $\mathcal{G}_D$ with root w.

As an example, the product of the description tree $\mathcal{G}_C$ shown in Figure 7 and the description tree $\mathcal{G}_{D'}$ shown in Figure 11 is depicted on the right-hand side of Figure 11. Thus, the lcs in $\mathcal{EL}$ of the concept descriptions

$$C := P \sqcap \exists r.(\exists r.(P \sqcap Q) \sqcap \exists s.Q) \sqcap \exists r.(P \sqcap \exists s.P) \quad \text{and} \quad D' := \exists r.(P \sqcap \exists r.P \sqcap \exists s.Q)$$

is $lcs_{\mathcal{EL}}(C, D') \equiv \exists r.(\exists r.P \sqcap \exists s.Q) \sqcap \exists r.(P \sqcap \exists s.\top)$.

This approach of computing the lcs as a product of description trees can be extended to $\mathcal{ALE}$ [25] and to $\mathcal{ALEN}$ [108]. The main difference is that the concept descriptions must be normalized appropriately before building the description trees.

Now, we come to the formal definition of the most specific concept. Let $\mathcal{L}$ be a DL. The $\mathcal{L}$ concept description E is the *most specific concept* (msc) in $\mathcal{L}$ of the individual a in the $\mathcal{L}$ ABox $\mathcal{A}$ ($msc_{\mathcal{L}}(a)$ for short) iff

1. $\mathcal{A} \models E(a)$, and
2. E is the least concept satisfying (i), i.e., if E' is an $\mathcal{L}$ concept description satisfying $\mathcal{A} \models E'(a)$ then $E \sqsubseteq E'$.

As with the lcs, the msc is unique up to equivalence, if it exists. In contrast to the lcs, which usually exists for standard DLs, the msc does not always exist in $\mathcal{EL}$, $\mathcal{ALN}$, and $\mathcal{ALE}$. This is due to the presence of so-called role cycles in the ABox. For example, w.r.t. the ABox

$$\{\mathsf{loves}(\mathsf{NARCIS}, \mathsf{NARCIS}),\ \mathsf{Vain}(\mathsf{NARCIS})\},$$

the individual NARCIS does not have an msc in $\mathcal{EL}$. In fact, assume that E is the msc of NARCIS. Then E has a finite role depth, i.e., a finite maximal number of nestings of existential restrictions. If this role depth is smaller than n, then E is not subsumed by the $\mathcal{EL}$ concept description

$$E' := \underbrace{\exists \mathsf{loves}. \cdots \exists \mathsf{loves}.}_{n \text{ times}} \mathsf{Vain},$$

in spite of the fact that NARCIS is an instance of E'. The same example works for $\mathcal{ALE}$, and a similar one can be given for $\mathcal{ALN}$ [20].

One way to overcome this problem is to allow for cyclic TBoxes interpreted with greatest fixpoint semantics. In the above example, the defined concept $\mathsf{Narcis} \equiv \mathsf{Vain} \sqcap \exists \mathsf{loves}.\mathsf{Narcis}$ is then an msc of the individual NARCIS. In order to employ this approach in the bottom-up construction of DL knowledge bases, one must allow these knowledge bases to contain cyclic definitions. Thus, also the subsumption problem and the problem of computing the lcs must be solved w.r.t. cyclic definitions interpreted with greatest fixpoint semantics. In [106, 20], this is done for $\mathcal{ALN}$, and in [9, 7] for $\mathcal{EL}$. The appropriate treatment of cyclic TBoxes in $\mathcal{ALE}$ is still an open problem.

Another possibility is to approximate the msc by restricting the attention to concept descriptions whose role depth is bounded by a fixed number k [53, 107].

5.3 *Matching and unification*

Concept patterns are concept descriptions in which *concept variables* (usually denoted by X, Y, etc.) may occur in place of concept names. The main difference between concept names and concept variables is that the latter can be replaced by concept descriptions when applying a substitution.

For example, $D := P \sqcap X \sqcap \forall r.(Y \sqcap \forall r.X)$ is a concept pattern containing the concept variables X and Y. By applying the substitution $\sigma := \{X \mapsto Q,\ Y \mapsto \forall r.P\}$ to it, we obtain the concept description

$$\sigma(D) = P \sqcap Q \sqcap \forall r.(\forall r.P \sqcap \forall r.Q).$$

Let $\mathcal{L}$ be a DL. An $\mathcal{L}$ *unification problem* is of the form

$$C_1 \equiv^? D_1, \; \ldots, \; C_n \equiv^? D_n,$$

where $C_1, \ldots, D_n$ are $\mathcal{L}$ concept patterns. A *unifier* of this problem is a substitution σ such that $\sigma(C_i) \equiv \sigma(D_i)$ for $i = 1, \ldots, n$.

For unification, the only results available until now are for the small DL $\mathcal{FL}_0$ and its extension $\mathcal{FL}_{reg}$ by the role constructors union, composition, and reflexive-transitive closure. In [28], it is shown that deciding unifiability of $\mathcal{FL}_0$-patterns is an ExpTime-complete problem, and in [22] this result is extended to $\mathcal{FL}_{reg}$ and it's extension with $\bot$. In the following, we sketch how the results for unification in $\mathcal{FL}_0$ can be obtained. As shown in [28], we can without loss of generality restrict the attention to unification problems consisting of a single equation $C \equiv^? D$. Using the language-based normal form of $\mathcal{FL}_0$ concept descriptions, we can write the patterns C, D in the form

$$\begin{aligned} C &\equiv \forall S_{0,1}.A_1 \sqcap \ldots \sqcap \forall S_{0,k}.A_k \sqcap \forall S_1.X_1 \sqcap \ldots \sqcap \forall S_n.X_n, \\ D &\equiv \forall T_{0,1}.A_1 \sqcap \ldots \sqcap \forall T_{0,k}.A_k \sqcap \forall T_1.X_1 \sqcap \ldots \sqcap \forall T_n.X_n, \end{aligned}$$

where $A_1, \ldots, A_k$ are the concept names and $X_1, \ldots, X_n$ the concept variables occurring in C, D, and $S_{0,i}, S_j, T_{0,i}, T_j$ $(i = 1, \ldots, k, j = 1, \ldots, n)$ are finite sets of words over the alphabet of all role names. In [28], it is shown that $C \equiv^? D$ has a unifier iff for all $i = 1, \ldots, k$, the *linear language equation*

$$S_{0,i} \cup S_1 X_{1,i} \cup \cdots \cup S_n X_{n,i} = T_{0,i} \cup T_1 X_{1,i} \cup \cdots \cup T_n X_{n,i}$$

has a solution, i.e., we can substitute the variables $X_{j,i}$ by finite languages such that the equation holds. Note that this is not a system of k equations that must be solved simultaneously: since they do not share variables, each of these equations can be solved separately.

Let us illustrate the connection between $\mathcal{FL}_0$ unification problems and linear language equations by a simple example. The normal forms of the concept patterns

$$C := \forall r.(A_1 \sqcap \forall r.A_2) \sqcap \forall r.\forall s.X_1 \quad \text{and} \quad D := \forall r.\forall s.(\forall s.A_1 \sqcap \forall r.A_2) \sqcap \forall r.X_1 \sqcap \forall r.\forall r.A_2$$

are

$$C \equiv \forall\{r\}.A_1 \sqcap \forall\{rr\}.A_2 \sqcap \forall\{rs\}.X_1 \quad \text{and} \quad D \equiv \forall\{rss\}.A_1 \sqcap \forall\{rsr, rr\}.A_2 \sqcap \forall\{r\}.X_1.$$

Thus, the unification problem $C \equiv^? D$ leads to the two linear language equations

$$\begin{aligned} \{r\} \cup \{rs\}X_{1,1} &= \{rss\} \cup \{r\}X_{1,1}, \\ \{rr\} \cup \{rs\}X_{1,2} &= \{rsr, rr\} \cup \{r\}X_{1,2}. \end{aligned}$$

The first equation (the one for A_1) has $X_{1,1} = \{\varepsilon, s\}$ as a solution, and the second (the one for A_2) has $X_{1,2} = \{r\}$ as a solution. These two solutions yield the following unifier of $C \equiv^? D$:

$$\{X_1 \mapsto A_1 \sqcap \forall s.A_1 \sqcap \forall r.A_2\}.$$

By an exponential time reduction to the emptiness problem of top-down automata on finite trees it is shown in [28] that solvability of linear language equations of the form introduced above can be decided in exponential time. ExpTime-hardness is shown by

a reduction from the intersection emptiness problem for deterministic top-down tree automata. This shows that solvability of $\mathcal{FL}_0$ unification problems is an ExpTime-complete problem. In [22], these results are extended to $\mathcal{FL}_{reg}$. Basically, instead of linear language equations over finite sets, one obtains linear language equations over regular sets, and uses automata working on infinite trees to solve them.

An extension of these results to more expressive DLs, such as $\mathcal{ALC}$, appears to be very hard. This is supported by the fact that research on unification in modal logics has also not yet produced results on unification in K. In modal logic, unification can be seen as a special case of testing for the admissibility of an inference rule (see Chapter 8 for more details and references). The rule

$$\frac{\alpha_1(x_1,\ldots,x_n),\ldots,\alpha_m(x_1,\ldots,x_n)}{\beta(x_1,\ldots,x_n)}$$

is called *admissible* in a logic $\mathcal{L}$ iff every substitution of the x_i by formulae making $\alpha_1,\ldots,\alpha_m$ valid also makes β valid. If $\mathcal{L}$ is consistent, then the rule $\alpha(x_1,\ldots,x_n)/\bot$ is admissible iff $\alpha(x_1,\ldots,x_n) \equiv^? \top$ is not unifiable. If the logic is propositionally closed, then all unification problems can be brought into this form. Consequently, decidability of the admissible inference rule problem (e.g., in $\mathsf{K4}$, Grz) implies decidability of the unification problem.

Kracht also shows in Chapter 8 that admissibility of inference rules can be reduced to unification if every unification problem has a computable finite complete set of unifiers (see [32] for the relevant definitions from unification theory). Using Ghilardi's results that unification in $\mathsf{K4}$, $\mathsf{S4}$, and intuitionistic logic is finitary in this sense [76, 75], this shows that admissibility of inference rules is decidable for intuitionistic logic. It should be noted however, that these result consider only elementary unification, i.e., unification without free constants. In the DL setting introduced above, this means that they do not allow for concept names in concept descriptions (only concept variables). Also note that unification in $\mathcal{FL}_0$ is not finitary [2]. For the modal logic K, decidability of both admissibility of inference rules and unification are open problems (and generally assumed to be very hard).

Matching can be seen as a special case of unification where the left-hand sides of the unification problem are concept descriptions, i.e., the concept descriptions C_i in $C_i \equiv^? D_i$ do not contain variables. For DLs that are propositionally closed, unification can be reduced to matching. Indeed, it is easy to see that the equation $C \equiv^? D$ has the same solutions as $\top \equiv^? (C \sqcap D) \sqcup (\neg C \sqcap \neg D)$. Thus, for $\mathcal{ALC}$, matching is as hard as unification. For sub-Boolean DLs, matching can be significantly easier than unification (see below).

In [39], a different notion of matching, called *matching modulo subsumption*, was introduced.[21] In this setting, a matching problem is of the form $C \sqsubseteq^? D$ where C is a concept description and D a concept pattern. A *matcher* is then a substitution σ such that $C \sqsubseteq \sigma(D)$. Since $C \sqsubseteq \sigma(D)$ iff $C \sqcap \sigma(D) \equiv C$, and $C \sqcap \sigma(D) = \sigma(C) \sqcap \sigma(D) = \sigma(C \sqcap D)$, this matching problem modulo subsumption can be reduced to the following matching problem modulo equivalence: $C \equiv^? C \sqcap D$.

However, in many cases, matching modulo subsumption is simpler than matching modulo equivalence since it can be reduced to the subsumption problem. This is the case

[21] In the following, we call matching problems of the form $C \equiv^? D$ matching problems *modulo equivalence* to distinguish them from matching problems modulo subsumption.

for DLs that allow for $\top$ and where all constructors are *monotonic*, i.e., replacing their arguments by larger ones w.r.t. subsumption yields a larger description. An example of a monotonic constructor is conjunction: if $C \sqsubseteq C'$ and $D \sqsubseteq D'$, then $C \sqcap D \sqsubseteq C' \sqcap D'$. Other examples are value and existential restrictions as well as disjunction. For such a DL, $C \sqsubseteq^? D$ has a matcher iff the substitution $\sigma_\top$ that replaces all variables by $\top$ is a matcher, i.e., if $C \sqsubseteq \sigma_\top(D)$. In fact, monotonicity of the constructors implies that $\sigma(D) \sqsubseteq \sigma_\top(D)$ holds for all substitutions σ, and thus whenever there is a matcher σ, then $\sigma_\top$ is also a matcher. Also note that matching cannot be simpler than subsumption since the matching problem $C \sqsubseteq^? D$ where D does not contain variables has a solution iff $C \sqsubseteq D$.

In the context of matching modulo subsumption, one is, however, usually not interested in arbitrary solution, and in particular not in the trivial largest one $\sigma_\top$, but rather in *minimal* ones, i.e., in matchers σ of $C \sqsubseteq^? D$ such that there does not exist another substitution δ such that $C \sqsubseteq \delta(D) \sqsubset \sigma(D)$ (see [39] for a motivation). Computing minimal matcher may again be harder than simply testing whether the trivial solution candidate $\sigma_\top$ is indeed a matcher.

In [28], the language-based approach for unification is used to show that solvability of matching problems modulo equivalence (and thus also modulo subsumption) in $\mathcal{FL}_0$ can be decided in polynomial time, and that minimal matchers of matching problems modulo subsumption are unique up to equivalence and can be computed in polynomial time. In [23], this result is extended to $\mathcal{ALN}$.[22] Matching in $\mathcal{EL}$ and $\mathcal{ALE}$ is considered in [21]. For both DLs, matching modulo equivalence is NP-complete. As explained above, the complexity of matching modulo subsumption coincides with the complexity of the subsumption problem, i.e., it is polynomial for $\mathcal{EL}$ and NP-complete for $\mathcal{ALE}$. In the following, we consider the complexity of matching modulo equivalence in $\mathcal{EL}$ and $\mathcal{ALE}$ in some more detail. NP-hardness of matching in $\mathcal{ALE}$ is an immediate consequence of NP-hardness of subsumption in $\mathcal{ALE}$.

NP*-hardness of matching modulo equivalence in* $\mathcal{EL}$ is shown in [21] by a reduction from SAT. Let $\phi = \varphi_1 \wedge \cdots \wedge \varphi_m$ be a propositional formula in conjunctive normal form and let $\{p_1, \ldots, p_n\}$ be the propositional variables of this problem. For these variables, we introduce the concept variables $\{X_1, \ldots, X_n, \overline{X}_1, \ldots, \overline{X}_n\}$. Furthermore, we need concept names A and B as well as role names $r_1, \ldots, r_n$ and $s_1, \ldots, s_m$. First, we specify a matching problem $C_n \equiv^? D_n$ that encodes the truth values of the n propositional variables:

$$\begin{array}{lcl} C_n & := & \exists r_1.A \sqcap \exists r_1.B \sqcap \ldots \sqcap \exists r_n.A \sqcap \exists r_n.B \\ D_n & := & \exists r_1.X_1 \sqcap \exists r_1.\overline{X}_1 \sqcap \ldots \sqcap \exists r_n.X_n \sqcap \exists r_n.\overline{X}_n. \end{array}$$

The matchers of this problem are exactly the substitutions that replace X_i by A and $\overline{X}_i$ by B (corresponding to $p_i = \mathsf{true}$), or vice versa (corresponding to $p_i = \mathsf{false}$).

In order to encode ϕ, we introduce a concept pattern D_{φ_i} for each clause φ_i. For example, if $\varphi_i = p_1 \vee \neg p_2 \vee p_3 \vee \neg p_4$, then $D_{\varphi_i} := X_1 \sqcap \overline{X}_2 \sqcap X_3 \sqcap \overline{X}_4 \sqcap B$. The whole formula is then represented by the matching problem $C_\phi \equiv^? D_\phi$, where

$$C_\phi := \exists s_1.(A \sqcap B) \sqcap \ldots \sqcap \exists s_m.(A \sqcap B) \quad \text{and} \quad D_\phi := \exists s_1.D_{\varphi_1} \sqcap \ldots \sqcap \exists s_m.D_{\varphi_m}.$$

This matching problem ensures that, among all the variables in D_{φ_i}, at least one must be replaced by A. This corresponds to the fact that, within one clause φ_i, there must

[22] In the presence of atomic negation one defines patterns such atomic negation may not be applied to variables, and thus atomic negation does not destroy the monotonicity property introduced above.

$$C := \mathsf{W} \sqcap \exists \mathsf{c}.(\mathsf{W} \sqcap \exists \mathsf{c}.(\mathsf{W} \sqcap \mathsf{D}) \sqcap \exists \mathsf{c}.(\mathsf{W} \sqcap \mathsf{P})) \sqcap \exists \mathsf{c}.(\mathsf{W} \sqcap \mathsf{D} \sqcap \exists \mathsf{c}.(\mathsf{W} \sqcap \mathsf{P}))$$

$$D := \mathsf{W} \sqcap \exists \mathsf{c}.(X \sqcap \exists \mathsf{c}.(\mathsf{W} \sqcap Y)) \sqcap \exists \mathsf{c}.(X \sqcap Y)$$

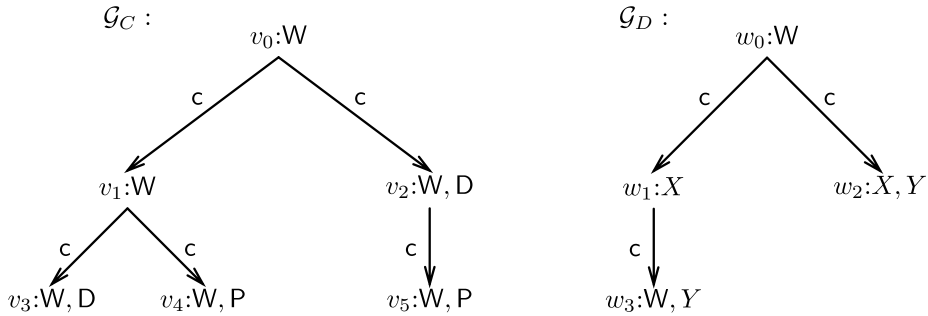


Figure 12. An $\mathcal{EL}$ concept description and an $\mathcal{EL}$ concept pattern, and the corresponding description trees.

be at least one literal that evaluates to true. Note that we need the concept B in D_{φ_i} to cover the case where all variables in D_{φ_i} are substituted with A. If we combine the two matching problems introduced above into a single problem $C_n \sqcap C_\phi \equiv^? D_n \sqcap D_\phi$, then it is easy to verify that ϕ is satisfiable iff this matching problem is solvable.

Membership in NP *for matching modulo equivalence in $\mathcal{EL}$ and $\mathcal{ALE}$* is an easy consequence of the following two (non-trivial) facts [21]. If an $\mathcal{EL}$ or $\mathcal{ALE}$ matching problem modulo equivalence has a matcher, then it has one of size polynomially bounded by the size of the problem. Furthermore, this matcher uses only concept and role names already contained in the matching problem. Thus, one can simply guess a substitution satisfying the given size bound, and then test (in P for $\mathcal{EL}$ and in NP for $\mathcal{ALE}$) whether it is a matcher.

Of course, this NP-algorithm for testing solvability of a matching problem does not yield a practical algorithm for actually computing matchers. A more practical *algorithm that computes all minimal matchers* of $\mathcal{EL}$ and $\mathcal{ALE}$ matching problems *modulo subsumption* is based on the characterization of subsumption through the existence of a homomorphism (i.e., a simulation relation that is a function) between the corresponding description trees [25]. As an example, consider the $\mathcal{EL}$ matching problem $C \sqsubseteq^? D$ for the concept description C and the concept pattern D depicted in Figure 12. Readers not liking such abstract examples may read W as Woman, D as Doctor, P as Professor, and c as child. Thus, the pattern describes concepts consisting of women that have (i) a child satisfying some property X and having a female child satisfying some property Y, and (ii) a child satisfying both X and Y.

When considering homomorphisms between the description trees of a concept pattern and a concept description, we simply ignore the concept variables, i.e., the inclusion condition between the labels does not take variables into account. In our example, there are six homomorphisms from $\mathcal{G}_D$ into $\mathcal{G}_C$. We consider the ones mapping w_i onto v_i for $i = 0, 1, 2$, and w_3 onto v_3 or w_3 onto v_4, which we denote by h_1 and h_2, respectively.

The matching algorithm described in [21] tries to construct substitutions τ such that $C \sqsubseteq \tau(D)$, i.e., there is a homomorphism from $\mathcal{G}_{\tau(D)}$ into $\mathcal{G}_C$. This is achieved by first computing all homomorphisms from $\mathcal{G}_D$ into $\mathcal{G}_C$. Assume that the node w in $\mathcal{G}_D$, whose label contains X, is mapped onto the note v of $\mathcal{G}_C$. The idea is then to substitute X with the concept description corresponding to the subtree of $\mathcal{G}_C$ starting with the node v. We will denote this description by C_v in the following. The remaining problem is that a variable X may occur more than once in D, and thus nodes containing X may be mapped to several nodes in $\mathcal{G}_C$. Thus, we cannot simply define $\tau(X)$ as $C_{h(w)}$ where w is such that X occurs in the label of w. Since there may exist several nodes w with this property, we take the least common subsumer of the corresponding parts of C. The reason for taking the *least* common subsumer is that we want to compute minimal matchers.

In our example, the homomorphism h_1 yields the substitution τ_1:

$$\tau_1(X) := \mathit{lcs}(C_{v_1}, C_{v_2}) \equiv \mathsf{W} \sqcap \exists \mathsf{c}.(\mathsf{W} \sqcap \mathsf{P}), \qquad \tau_1(Y) := \mathit{lcs}(C_{v_2}, C_{v_3}) \equiv \mathsf{W} \sqcap \mathsf{D},$$

whereas h_2 yields the substitution τ_2:

$$\tau_2(X) := \mathit{lcs}(C_{v_1}, C_{v_2}) \equiv \mathsf{W} \sqcap \exists \mathsf{c}.(\mathsf{W} \sqcap \mathsf{P}), \qquad \tau_2(Y) := \mathit{lcs}(C_{v_2}, C_{v_4}) \equiv \mathsf{W}.$$

The algorithm is guaranteed to compute all minimal matchers, but may also compute some non-minimal ones, which must be removed in a post-processing step. In our example, the substitution τ_1 is a minimal matcher, but τ_2 is not minimal. In general, a given matching problem modulo subsumption may have exponentially many inequivalent minimal matchers, and the size of these minimal matchers may also be exponential in the size of the matching problem [21].

5.4 Rewriting and approximation

In [26], a very general framework for rewriting in DLs is introduced, which has several interesting instances. In order to introduce this framework, we fix a set N_R of role names and a set N_P of primitive concept names. Now, let $\mathcal{L}_s$, $\mathcal{L}_d$, and $\mathcal{L}_t$ be three DLs (the source-, destination, and TBox-DL, respectively). A *rewriting problem* is given by

- an $\mathcal{L}_t$ TBox $\mathcal{T}$ containing only role names from N_R and primitive concepts from N_P; the set of defined concepts occurring in $\mathcal{T}$ is denoted by N_D;
- an $\mathcal{L}_s$ concept description C using only the names from N_R and N_P;
- a binary relation ρ between $\mathcal{L}_s$ concept descriptions and $\mathcal{L}_d$ concept descriptions.

An $\mathcal{L}_d$ *rewriting of* C *using* $\mathcal{T}$ is an $\mathcal{L}_d$ concept description E built using role names from N_R and concept names from $N_P \cup N_D$ such that $C \rho E$. Given an appropriate ordering $\preceq$ on $\mathcal{L}_d$ concept descriptions, a rewriting E is called $\preceq$-*minimal* iff there does not exist a rewriting E' such that $E' \prec E$.

To illustrate the use of this general framework by examples, we consider two of its instances in more detail: the *minimal rewriting problem* and the *approximation problem.*

Minimal rewriting

This is the instance of the framework where (i) all three DLs are the same language $\mathcal{L}$; (ii) the TBox $\mathcal{T}$ is acyclic; (iii) the binary relation ρ corresponds to equivalence w.r.t. the TBox; and (iv) $\mathcal{L}$ concept descriptions are ordered by size, i.e., $E \preceq E'$ iff $|E| \leq |E'|$. The size $|E|$ of a concept description E is defined to be the number of occurrences of concept and role names in E.

In order to determine the complexity of the minimal rewriting problem, Baader et al. [26] first analyse the *decision problem* induced by this optimization problem for a given DL $\mathcal{L}$: given an $\mathcal{L}$ concept description C, an acyclic $\mathcal{L}$ TBox $\mathcal{T}$, and a nonnegative integer κ, does there exist an $\mathcal{L}$ rewriting E of C using $\mathcal{T}$ such that $|E| \leq \kappa$? Since this decision problem can obviously be reduced to the problem of computing a minimal rewriting of C using $\mathcal{T}$, hardness results for the decision problem carry over to the optimization problem.

For $\mathcal{ALC}$, this decision problem is PSPACE-hard since the PSPACE-complete subsumption problem can be reduced to it. Indeed, let C, D be two $\mathcal{ALC}$ concept descriptions, and A, P_1, P_2 three different concept names not occurring in C, D. Then $C \sqsubseteq D$ iff there exists a minimal rewriting of size 1 of the $\mathcal{ALC}$ concept description $P_1 \sqcap P_2 \sqcap C$ using the TBox $\mathcal{T} := \{A \equiv P_1 \sqcap P_2 \sqcap C \sqcap D\}$ [26]. The two concept names P_1 and P_2 are introduced to ensure that the size of the concept description to be rewritten is strictly larger than the size of A.

However, subsumption is not the only source of complexity for the minimal rewriting problem. In fact, even for the small DL $\mathcal{FL}_0$, for which subsumption of concept descriptions and w.r.t. expanded TBoxes is polynomial, the rewriting problem (using an expanded TBox) is NP-hard. This is shown in [26] by a reduction from the set cover problem.

For an arbitrary DL $\mathcal{L}$, the minimal rewriting decision problem can obviously be decided by a non-deterministic polynomial time algorithm that uses an oracle for subsumption. This algorithm just guesses an $\mathcal{L}$ concept description over the available vocabulary and of size at most κ, and then checks whether this description is equivalent to the input description modulo the TBox. For $\mathcal{ALC}$, this shows that the minimal rewriting decision problem is PSPACE-complete. It can also be used to show that the problem is NP-complete for $\mathcal{FL}_0$ (see [26] for details).

Let us now come to the problem of actually *computing minimal rewritings*. The hardness results mentioned above imply that computing one minimal rewriting is already a hard problem. In addition, the following simple example shows that the number of minimal rewritings of a concept description C using a TBox $\mathcal{T}$ can be exponential in the size of C and $\mathcal{T}$.

For a nonnegative integer n, let $C_n := P_1 \sqcap \ldots \sqcap P_n$ and $\mathcal{T}_n := \{A_i \equiv P_i \mid 1 \leq i \leq n\}$. For each vector $\mathbf{i} = (i_1, \ldots, i_n) \in \{0,1\}^n$, we define

$$E_{\mathbf{i}} := \bigsqcap_{1 \leq j \leq n, i_j = 0} P_j \sqcap \bigsqcap_{1 \leq j \leq n, i_j = 1} A_j.$$

Obviously, for all $\mathbf{i} \in \{0,1\}^n$, $E_{\mathbf{i}}$ is a rewriting of C_n of size $|E_{\mathbf{i}}| = n = |C_n|$. Furthermore, it is easy to see that there does not exist a smaller rewriting of C_n using $\mathcal{T}_n$. Hence, there exists an exponential number of different minimal rewritings of C_n using $\mathcal{T}_n$.

A *naïve algorithm* for computing one minimal rewriting would enumerate all concept descriptions E of size $k = 1$, then $k = 2$, etc., until a rewriting E_0 of C using $\mathcal{T}$ is

encountered. By construction, this rewriting is minimal, and since C is a rewriting of itself, one need not consider sizes larger than $|C|$. If one is interested in computing all minimal rewritings, it remains to enumerate all concept descriptions of size $|E_0|$, and test for each of them whether they are equivalent to C modulo $\mathcal{T}$. Obviously, this algorithm is not practical.

The main ideas underlying the more practical algorithm described in [26] is the following. For a given input description C, one splits the computation of rewritings into two steps:

- Compute an *extension* C^* of C. Such an extension is obtained from C by conjoining defined concepts at some positions of C while making sure that $C \equiv_{\mathcal{T}} C^*$ holds.
- Compute a *reduction* $\widehat{C}$ of C^*. Such a reduction is obtained from C^* by removing certain parts of C^* while making sure that $C^* \equiv_{\mathcal{T}} \widehat{C}$ holds.

The exact definitions of the right notions of extension and reduction depend, of course, on the DL under consideration. In [26], these definitions are given for $\mathcal{ALE}$. It is shown that the algorithm obtained this way computes only writings of the input description, and that all minimal rewritings are among the computed rewritings. In addition, [26] describes a more efficient heuristic algorithm that is not guaranteed to find *minimal* rewritings, but behaves quite well in practice. Basically, this algorithm uses a greedy strategy in the extension step, i.e., it conjoins as many defined concepts as possible to each position of C.

Approximation

This is the instance of the framework where (i) $\mathcal{T}$ is empty, and thus $\mathcal{L}_t$ is irrelevant; (ii) both ρ and $\preceq$ are the subsumption relation $\sqsubseteq$. In this case, we talk about approximation rather than rewriting. Given two DLs $\mathcal{L}_s$ and $\mathcal{L}_d$, an $\mathcal{L}_d$ *approximation* of an $\mathcal{L}_s$ concept description C is thus an $\mathcal{L}_d$ concept description D such that $C \sqsubseteq D$ and D is minimal (w.r.t. subsumption) with this property.

The case where $\mathcal{L}_s = \mathcal{ALC}$ and $\mathcal{L}_d = \mathcal{ALE}$ is investigated in [48]. Recall that the only difference between $\mathcal{ALC}$ and $\mathcal{ALE}$ is that disjunction is disallowed in $\mathcal{ALE}$ concept descriptions.[23] If C_1, C_2 are $\mathcal{ALE}$ concept descriptions, then it is easy to see that the approximation of the $\mathcal{ALC}$ concept description $C_1 \sqcup C_2$ by an $\mathcal{ALE}$ concept description is $lcs_{\mathcal{ALE}}(C_1, C_2)$. This suggests the following approach for approximating an $\mathcal{ALC}$ concept description C by an $\mathcal{ALE}$ concept description: just replace every disjunction in C by an application of the lcs operation. The following example demonstrates that this approach is too naïve: let $C := (\forall r.B \sqcup (\exists r.B \sqcap \forall r.A)) \sqcap \exists r.A$. If we replace the disjunction by an lcs operation and then compute the lcs, we obtain the $\mathcal{ALE}$ concept description

$$lcs_{\mathcal{ALE}}(\forall r.B, (\exists r.B \sqcap \forall r.A)) \sqcap \exists r.A \equiv \top \sqcap \exists r.A \equiv \exists r.A.$$

However, this concept description is too general. It is easy to see that $C \sqsubseteq \exists r.(A \sqcap B) \sqsubset \exists r.A$. In fact, $\exists r.(A \sqcap B)$ is the correct approximation.

In order to overcome this problem, the $\mathcal{ALC}$ concept description has to be transformed into an appropriate normal form. Basically, this normal form is obtained by distributing

[23] Here we assume without loss of generality that all $\mathcal{ALC}$ concept descriptions are in negation normal form where negation occurs only in front of concept names.

conjunctions over disjunctions, and by applying the rule $\forall r.C \sqcap \forall r.D \to \forall r.(C \sqcap D)$. For the example from above, the normal form is

$$C \equiv (\forall r.B \sqcap \exists r.A) \sqcup (\exists r.B \sqcap \forall r.A \sqcap \exists r.A),$$

and $lcs_{\mathcal{ALE}}(\forall r.B \sqcap \exists r.A, \exists r.B \sqcap \forall r.A \sqcap \exists r.A) = \exists r.(A \sqcap B)$.

However, even for $\mathcal{ALC}$ concept descriptions in this normal form, one cannot simply replace disjunction by the lcs operation to obtain their $\mathcal{ALE}$ approximation. Consider the $\mathcal{ALC}$ concept description $C' = \exists r.A \sqcap \exists r.B \sqcap \forall r.(\neg A \sqcup \neg B)$. If we simply replace the disjunction by the lcs, then we obtain $\exists r.A \sqcap \exists r.B \sqcap \forall r.\top \equiv \exists r.A \sqcap \exists r.B$. However, C' is also subsumed by the more specific $\mathcal{ALE}$ concept description $\exists r.(A \sqcap \neg B) \sqcap \exists r.(B \sqcap \neg A)$. This problem can be overcome by also propagating value restrictions onto existential restrictions. An approximation algorithms based on these ideas is described in [48]. It is shown that every $\mathcal{ALC}$ concept description has an $\mathcal{ALE}$ approximation, and this approximation is unique up to equivalence, i.e., there is always a least approximation. However, the size of the approximation may grow exponentially with the size of the input description. The algorithm for computing the approximation given in [48] runs in doubly-exponential time, and it is not clear whether this time bound can be improved. In [47], these results are extended to the approximation of $\mathcal{ALCN}$ concepts descriptions by $\mathcal{ALEN}$ concept descriptions.

6 NON-STANDARD EXPRESSIVITY

As discussed in Section 2, many expressive means of description logics have a counterpart in modal logic. In this section, we discuss two expressive means that are important for DLs, but lack a direct modal counterpart: concrete domains and role value maps.

6.1 Concrete Domains

The purpose of concrete domains is to enable the definition of concept descriptions with reference to concrete qualities of real-world objects such as their age, weight, temperature, and spatial extension. For example, we may define a teenager as a human whose age is between 10 and 19, or formulate a GCI stating that the age of a child is always smaller than the age of its parents. Representing concrete qualities and constraints of this form is necessary in almost all applications of description logics, such as reasoning about the semantic web [19] and about conceptual database models [114]. For this reason, even early DL systems such as MESON [68] and CLASSIC [44] addressed the issue of representing concrete qualities. However, these early approaches were of a rather ad hoc nature. The first approach that was fully (and formally) integrated with a description logic was presented by Baader and Hanschke [14], who proposed to extend the description logic $\mathcal{ALC}$ with so-called concrete domains.

Definitions

A *concrete domain* $\mathcal{D}$ is a pair $(\Delta^{\mathcal{D}}, \Phi^{\mathcal{D}})$ consisting of a non-empty set $\Delta^{\mathcal{D}}$ and a collection $\Phi^{\mathcal{D}}$ of *predicates names* such that each predicate $P \in \Phi^{\mathcal{D}}$ is equipped with an arity n and a fixed extension $P^{\mathcal{D}} \subseteq (\Delta^{\mathcal{D}})^n$. Slightly abusing notation, we will sometimes refer to the set $\Delta^{\mathcal{D}}$ as the concrete domain. In contrast, the domain $\Delta^{\mathcal{I}}$ of interpretations $\mathcal{I}$ will

be called the *abstract domain.* For many application areas, the most interesting concrete domains are numerical ones. A typical numerical concrete domain is $\mathsf{Q} = (\mathbb{Q}, \Phi^{\mathsf{Q}})$, where $\mathbb{Q}$ denotes the rational numbers, and Φ^{Q} is comprised of the following predicates:

- unary predicates P_q for each $P \in \{<, \leq, =, \neq, \geq, >\}$ and each $q \in \mathbb{Q}$ with $(P_q)^{\mathsf{Q}} = \{q' \in \mathbb{Q} \mid q' \; P \; q\}$;
- binary predicates $<, \leq, =, \neq, \geq, >$ with the obvious extensions;
- a ternary predicate $+$ with $(+)^{\mathsf{Q}} = \{(q, q', q'') \in \mathbb{Q}^3 \mid q + q' = q''\}$.

By integrating a concrete domain $\mathcal{D}$ into $\mathcal{ALC}$, we obtain the basic description logic with concrete domains $\mathcal{ALC}(\mathcal{D})$. More precisely, $\mathcal{ALC}(\mathcal{D})$ is obtained from $\mathcal{ALC}$ by augmenting it with

- *abstract features*: a new sort of roles that is interpreted as a partial function from $\Delta^{\mathcal{I}}$ to $\Delta^{\mathcal{I}}$; abstract features can be used inside value restrictions and existential restrictions;
- *concrete features*: a new sort of roles that is interpreted as a partial function from the abstract domain $\Delta^{\mathcal{I}}$ into the concrete domain $\Delta^{\mathcal{D}}$; concrete features can *not* be used inside value restrictions and existential restrictions;
- a new concept constructor $P(u_1, \ldots, u_n)$, where $P \in \Phi^{\mathcal{D}}$ is a predicate of arity n, and each u_i is an expression $f_1 \circ \cdots \circ f_k \circ g$ with $f_1, \ldots, f_k$ $(k \geq 0)$ abstract features and g a concrete feature. In the following, such expressions will be called *concrete paths.* The semantics of the new constructor is

$$P(u_1, \ldots, u_n)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \exists x_1, \ldots, x_n \in \Delta^{\mathcal{D}} : u_i^{\mathcal{I}}(d) = x_i \text{ for } 1 \leq i \leq n \text{ and } (x_1, \ldots, x_n) \in P^{\mathcal{D}}\},$$

  where the interpretation $u^{\mathcal{I}}$ of a concrete path $u = f_1 \circ \cdots \circ f_k \circ g$ is defined as the partial function that maps $d \in \Delta^{\mathcal{I}}$ to $g^{\mathcal{I}}(f_k^{\mathcal{I}} \cdots (f_1^{\mathcal{I}}(d)) \cdots)$.

Using the concrete domain Q, the teenagers mentioned above can now be defined as

$$\mathsf{Teenager} \equiv \mathsf{Human} \sqcap >_9(\mathsf{age}) \sqcap <_{20}(\mathsf{age})$$

where age is a concrete feature. Similarly, the constraint saying that the age of children is smaller than the age of their parents can be formulated as

$$\top \sqsubseteq <(\mathsf{age}, \mathsf{mother} \circ \mathsf{age}) \sqcap <(\mathsf{age}, \mathsf{father} \circ \mathsf{age}),$$

where mother and father are abstract features.

There is a slight difference between the logic $\mathcal{ALC}(\mathcal{D})$ as defined here and the original version introduced in [14]: Baader and Hanschke's variant uses only a single type of feature whose interpretation is a partial function from $\Delta^{\mathcal{I}}$ to $\Delta^{\mathcal{I}} \cup \Delta^{\mathcal{D}}$. Thus, this type of feature combines our abstract and concrete features into one sort. In the literature, both versions of $\mathcal{ALC}(\mathcal{D})$ are considered. All results discussed in this section hold for both versions. Also note that the assumption that $\mathcal{ALC}$ is extended with only *one* concrete domain can be made without loss of generality, as it is shown in [14] that multiple concrete domains can be combined into a single one. In the world of modal logic, the closest relatives to DLs with concrete domains are linear temporal logics with constraints, see for example [33, 60].

Basic Results

When considering a description logic that is equipped with concrete domains, it is most desirable to obtain decidability results and complexity bounds that do not depend on a particular concrete domain, but rather apply to a class of concrete domains that is as large as possible. The first (decidability) result in this spirit was given by Baader and Hanschke in their original paper. Their result concerns the satisfiability of $\mathcal{ALC}(\mathcal{D})$ concept descriptions, where the concrete domain $\mathcal{D}$ is only required to satisfy some weak conditions. These conditions are derived from the fact that any satisfiability algorithm not committing itself to a particular concrete domain must call some concrete domain reasoner as a subprocedure via a well-defined interface. This observation leads to the notion of *admissibility*.

Let $\mathcal{D}$ be a concrete domain and V a countably infinite set of variables. A $\mathcal{D}$-*conjunction* is a predicate conjunction of the form

$$c = \bigwedge_{i<k} P_i(x_0^{(i)}, \dots, x_{n_i}^{(i)})$$

where $P_i \in \Phi^{\mathcal{D}}$ is an n_i-ary predicate for each $i < k$ and the $x_j^{(i)}$ are variables from V. A $\mathcal{D}$-conjunction c is *satisfiable* iff there exists a function δ mapping the variables in c to elements of $\Delta^{\mathcal{D}}$ such that $(\delta(x_0^{(i)}), \dots, \delta(x_{n_i}^{(i)})) \in P_i^{\mathcal{D}}$ for each $i < k$. We say that the concrete domain $\mathcal{D}$ is *admissible* iff

1. its set of predicates is closed under negation[24] and contains a name $\top_{\mathcal{D}}$ for $\Delta^{\mathcal{D}}$, and

2. the satisfiability of $\mathcal{D}$-conjunctions is decidable.

We refer to the satisfiability of $\mathcal{D}$-conjunctions as $\mathcal{D}$-*satisfiability*. Property 1 of admissibility has to be satisfied since $\mathcal{ALC}(\mathcal{D})$ provides for negation: for example, the concept description $C := {=}(g_1, g_1)) \sqcap {=}(g_2, g_2) \sqcap \neg{<}(g_1, g_2)$ is such that $d \in C^{\mathcal{I}}$ implies $g_1^{\mathcal{I}}(d) \geq g_2^{\mathcal{I}}(d)$ without explicitly using the "$\geq$" predicate,[25] and such information must be conveyed to the concrete domain reasoner. Note that the concrete domain Q presented above can easily be extended to satisfy Property 1 of admissibility: simply add predicates $\top_{\mathsf{Q}}$, $\bot_{\mathsf{Q}}$, and $\overline{+}$ (i.e., the negation of "+") with the obvious extensions. Let Q^a denote the extended version of Q. By using a reduction to linear programming, it is straightforward to show that Q^a-satisfiability is decidable in polynomial time [115], and thus Q^a is admissible.

The basic decidability result for $\mathcal{ALC}(\mathcal{D})$ given by Baader and Hanschke states that satisfiability (and thus also subsumption) of $\mathcal{ALC}(\mathcal{D})$-concept descriptions is decidable if $\mathcal{D}$ is admissible [14]. The complexity of this problem has been analyzed by Lutz [113], who proved PSpace-completeness under the assumption that $\mathcal{D}$ is admissible and $\mathcal{D}$-satisfiability is in PSpace. Thus if $\mathcal{D}$-satisfiability is in PSpace, then adding concrete domains to $\mathcal{ALC}$ does not increase the complexity of reasoning. Since Q^a-satisfiability is a polynomial time problem, we obtain PSpace-completeness for the instance $\mathcal{ALC}(\mathsf{Q}^a)$ of $\mathcal{ALC}(\mathcal{D})$. A discussion of the complexity of $\mathcal{D}$-satisfiability for a variety of numerical, temporal, and spatial concrete domains can be found in [115, 112].

---

[24] i.e., for each $P \in \Phi^{\mathcal{D}}$ of arity n, we find a $\overline{P} \in \Phi^{\mathcal{D}}$ with $\overline{P}^{\mathcal{D}} = (\Delta^{\mathcal{D}})^n \setminus P^{\mathcal{D}}$.

[25] The first two conjuncts are needed to ensure that $g_1^{\mathcal{I}}(d)$ and $g_2^{\mathcal{I}}(d)$ are actually defined.

When TBoxes are admitted, the complexity of reasoning increases drastically. We first consider acyclic TBoxes. As discussed in Section 2.2, satisfiability and subsumption w.r.t. acyclic TBoxes can be reduced to satisfiability w.r.t. the empty TBox using expansion. Thus, we obtain decidability of $\mathcal{ALC}(\mathcal{D})$ reasoning w.r.t. acyclic TBoxes if $\mathcal{D}$ is admissible. Since expansion is worst-case exponential, we also obtain an ExpSpace upper bound if $\mathcal{D}$-satisfiability is in PSpace. In the case of $\mathcal{ALC}$ without concrete domains, this upper bound can be improved to a PSpace one. Quite surprisingly, we can only push down the ExpSpace upper bound to a NExpTime one in the case of $\mathcal{ALC}(\mathcal{D})$: as proved in [117], there exists a concrete domain $\mathcal{D}$ such that $\mathcal{D}$-satisfiability is in PTime and satisfiability in $\mathcal{ALC}(\mathcal{D})$ w.r.t. acyclic TBoxes is NExpTime-hard. A matching upper bound states that satisfiability in $\mathcal{ALC}(\mathcal{D})$ w.r.t. acyclic TBoxes is in NExpTime if $\mathcal{D}$ is admissible and $\mathcal{D}$-satisfiability is in NP [117]. For subsumption, this yields analogous co-NExpTime bounds.

The jump in complexity from PSpace-complete to NExpTime-complete that is induced by adding acyclic TBoxes to $\mathcal{ALC}(\mathcal{D})$ is due to the fact that this addition increases the succinctness of $\mathcal{ALC}(\mathcal{D})$ (but not the expressivity). The NExpTime lower bound has been proved by reduction of a NExpTime-complete variant of the Post Correspondence Problem (PCP). As we cannot describe the reduction in full detail here, we sketch only how it makes use of the succinctness of acyclic TBoxes. The key observation is that it is possible to devise an acyclic TBox of size $O(k)$ that enforces models (of the concept name L_0) to contain a binary tree of depth k such that left successors are reachable via the abstract feature ℓ and right successors are reachable via the abstract feature r:

$$L_0 \equiv \exists \ell.L_1 \sqcap \exists r.L_1, \quad \ldots, \quad L_{k-1} \equiv \exists \ell.L_k \sqcap \exists r.L_k.$$

Without TBoxes, such a tree can only be enforced with a concept of length exponential in k. For the reduction, we add concept definitions expressing that the (exponentially many) leaves of the tree are connected via a chain of concrete domain predicates. For example, if we augment the above TBox with the following concept definitions and consider models of the conjunction $L_0 \sqcap C_0$, then we enforce that each leaf has a smaller number stored in the concrete feature g than all leaves that are to the right of it:

$$\begin{array}{rcl} C_0 & \equiv & <(\ell r^{k-1} g, r\ell^{k-1} g) \sqcap \forall \ell.C_1 \sqcap \forall r.C_1, \\ & \vdots & \\ C_{k-2} & \equiv & <(\ell r g, r \ell g) \sqcap \forall \ell.C_{k-1} \sqcap \forall r.C_{k-1}, \\ C_{k-1} & \equiv & <(\ell g, r g). \end{array}$$

Intuitively, the exponentially long chain of concrete domain predicates connecting the leaves can now be used to simulate the exponentially time-bounded computation of a Turing machine, or to talk about the concatenation of words in a PCP.

We now consider reasoning in $\mathcal{ALC}(\mathcal{D})$ with respect to GCIs, starting with a closely related result: let $\mathcal{ALC}^+(\mathcal{D})$ be the extension of $\mathcal{ALC}(\mathcal{D})$ with a transitive closure operator on roles and abstract features. Baader and Hanschke [16] prove that reasoning in $\mathcal{ALC}^+(\mathsf{R})$ w.r.t. the empty TBox is undecidable, where R is the concrete domain of real numbers with predicates based on Tarski algebra [149]. Their proof can easily be adapted to reasoning in $\mathcal{ALC}(\mathsf{R})$ w.r.t. GCIs, which is thus also undecidable. This adaptation is performed in [117], where a more general result is obtained: satisfiability (and thus also subsumption) in $\mathcal{ALC}(\mathcal{D})$ w.r.t. GCIs is undecidable if the concrete domain $\mathcal{D}$ satisfies

$\mathbb{N} \subseteq \Delta^{\mathcal{D}}$, and $\Phi^{\mathcal{D}}$ provides for a unary predicate for equality with 0, a binary equality predicate, and a binary predicate for incrementation. Thus, reasoning in $\mathcal{ALC}(\mathsf{Q})$ w.r.t. GCIs is undecidable since, in Q, incrementation can be expressed using the predicates "$=_1$" and "$+$". There are two ways for overcoming this rather disappointing result: either use a less powerful concrete domain constructor or very carefully choose the concrete domain.

The first approach was adopted, among others, by Möller et al. [80] and by Horrocks and Sattler [95, 130]. The imposed restriction on the concrete domain constructor usually is to allow only concrete features inside the concrete domain constructor instead of concrete paths of arbitrary length. In the following, the variant of $\mathcal{ALC}(\mathcal{D})$ obtained by this restriction will be called *path-free*. A particular form of path-freeness is to admit only unary predicates as proposed in [95]: in this case, reasoning in $\mathcal{ALC}(\mathcal{D})$ can be reduced to reasoning in path-free $\mathcal{ALC}(\mathcal{D})$ by replacing each concept description $P(f_1 \circ \cdots \circ f_k \circ g)$ with the equivalent path-free one $\exists f_1.\exists f_2. \cdots .\exists f_k.P(g)$. In [78] and [130], it is shown that reasoning in $\mathcal{SHN}(\mathcal{D})$ and $\mathcal{SHOQ}(\mathcal{D})$, the extensions of two expressive fragments of $\mathcal{SHOIQ}$ by concrete domains, is decidable w.r.t. GCIs if path-freeness is assumed and the concrete domain $\mathcal{D}$ is admissible. A more general result has been obtained in Section 5.3 of [27], where it is shown that any description logic $\mathcal{L}$ such that (i) satisfiability in $\mathcal{L}$ w.r.t. GCIs is decidable and (ii) $\mathcal{L}$'s class of interpretations is closed under disjoint unions (see [27] for details) can be extended with the path-free variant of the concrete domain constructor without losing decidability—provided that the concrete domain is admissible. Indeed, the "harmlessness" of the path-free concrete domain constructor is not very surprising since dropping concrete paths deprives concrete domains of most of their expressive power. For this reason, the complexity of reasoning w.r.t. GCIs in a DL incorporating path-free concrete domains is often not harder than the corresponding problem without concrete domains (if it dominates the complexity of $\mathcal{D}$-satisfiability). For example, in Section 2.4.1 of [112], it is shown that satisfiability in path-free $\mathcal{ALC}(\mathcal{D})$ w.r.t. GCIs is ExpTime-complete if $\mathcal{D}$ is admissible and $\mathcal{D}$-satisfiability is in ExpTime.

The second approach to overcome undecidability of $\mathcal{ALC}(\mathcal{D})$ with GCIs is to keep the original version of the concrete domain constructor and identify concrete domains that do not destroy decidability of reasoning if combined with GCIs. The first positive result following this route was established in [116], where a concrete domain C (for *c*omparison) is considered that is based on the rational numbers $\mathbb{Q} = \Delta^{\mathsf{C}}$, and provides for the binary predicates $<, \leq, =, \neq, \geq$, and $>$. It is shown that satisfiability (and thus also subsumption) in $\mathcal{ALC}(\mathsf{C})$ w.r.t. GCIs is ExpTime-complete, and that an analogous result holds for an interval-based temporal concrete domain. In [111], these results are further improved: first, the concrete domain C is extended to C^+, which additionally admits unary predicates $=_q$ for each $q \in \mathbb{Q}$; and second, the description logic is extended from $\mathcal{ALC}(\mathcal{D})$ to $\mathcal{SHIQ}(\mathcal{D})$, i.e. $\mathcal{SHOIQ}$ without nominals, but with the concrete domain C^+. For this extended logic, an ExpTime result analogous to the one stated above is established. A more general result is proved in [120], where a property of concrete domains is identified that is sufficient for decidability of $\mathcal{ALC}(\mathcal{D})$ with GCIs: a concrete domain $\mathcal{D}$ is called *ω-admissible* if it satisfies all of the following:

- $\mathcal{D}$ has only binary predicates;
- $\mathcal{D}$ has compactness: an infinite $\mathcal{D}$-conjunction is satisfiable if and only if every finite sub-conjunction is satisfiable;

- $\mathcal{D}$ has the patchwork property, which is defined as follows. A $\mathcal{D}$-conjunction c is *complete* iff, for all variables x, y occurring in c, c contains exactly one conjunct $P(x, y)$ and exactly one conjunct $P(y, x)$. Then, $\mathcal{D}$ has the *patchwork property* if the union of two satisfiable and complete $\mathcal{D}$-conjunctions that agree w.r.t. the conjuncts $P(x, y)$ and $P(y, x)$ w.r.t. all shared variables x, y is satisfiable.

The concrete domain C and the temporal concrete domain based on time intervals considered in [116] are ω-admissible. Additionally, it is shown in [120] that a spatial concrete domain based on the topological RCC8 relations is also ω-admissible, and therefore the corresponding incarnation of $\mathcal{ALC}(\mathcal{D})$ is decidable with GCIs. The result in [120] for DLs with ω-admissible concrete domains is established using a tableau algorithm and does not yield tight upper complexity bounds.

Roles in the Concrete Domain Constructor

The reader may wonder why the concrete domain constructor is introduced along with abstract features, instead of admitting normal roles in concrete paths. Indeed, this variant of concrete domains has also been considered by Hanschke [82]: a *concrete role path* is an expression $r_1 \circ \cdots \circ r_k \circ g$ with $r_1, \ldots, r_k$ roles and g a concrete feature. Then we may extend $\mathcal{ALC}$ with the concept constructors $\forall R_1, \ldots, R_k.P$ and $\exists R_1, \ldots, R_k.P$, where $R_1, \ldots, R_k$ are concrete role paths and $P \in \Phi^{\mathcal{D}}$ is a predicate of arity k. The semantics of these constructors is as follows:

$$\begin{array}{rcl}
(\forall R_1, \ldots, R_k.P)^{\mathcal{I}} & := & \{d \in \Delta^{\mathcal{I}} \mid \forall x_1, \ldots, x_k \in \Delta^{\mathcal{D}} : \\
& & \qquad (d, x_i) \in R_i^{\mathcal{I}} \text{ for } 1 \leq i \leq k \text{ implies } (x_1, \ldots, x_k) \in P^{\mathcal{D}}\} \\
(\exists R_1, \ldots, R_k.P)^{\mathcal{I}} & := & \{d \in \Delta^{\mathcal{I}} \mid \exists x_1, \ldots, x_k \in \Delta^{\mathcal{D}} : \\
& & \qquad (d, x_i) \in R_i^{\mathcal{I}} \text{ for } 1 \leq i \leq k \text{ and } (x_1, \ldots, x_k) \in P^{\mathcal{D}}\}
\end{array}$$

where the interpretation $R^{\mathcal{I}}$ of concrete role paths R is defined in the obvious way through relational composition.[26] The resulting DL is called $\mathcal{ALCP}(\mathcal{D})$. Reasoning with $\mathcal{ALCP}(\mathcal{D})$ concept descriptions has been proved to be decidable in [82]. When investigating the complexity of $\mathcal{ALCP}(\mathcal{D})$, it becomes clear that the restriction to abstract features inside the concrete domain constructor has computational advantages: it is shown in [117] that there exists a concrete domain $\mathcal{D}$ such that $\mathcal{D}$-satisfiability is in PTime and satisfiability of $\mathcal{ALCP}(\mathcal{D})$ concept descriptions is NExpTime-hard. Again, a matching upper bound is obtained for the case where $\mathcal{D}$-satisfiability is in NP. This should be contrasted with the PSpace-completeness of satisfiability of concept descriptions in $\mathcal{ALC}(\mathcal{D})$.

We have seen that both the generalized concrete domain constructor and acyclic TBoxes are seemingly moderate extension of $\mathcal{ALC}(\mathcal{D})$ that make reasoning considerably harder. Other such extensions include inverse roles, role conjunction, nominals, and a concrete domain *role* constructor [117, 118]. Thus, the PSpace upper bound of $\mathcal{ALC}(\mathcal{D})$ is not robust w.r.t. extensions of the language.

Uniqueness Constraints and Functional Dependencies

Uniqueness constraints (sometimes also called *identification constraints* and *keys*) and functional dependencies play an important role in the database area, and are also useful

[26] In $\mathcal{ALC}(\mathcal{D})$ with only functional roles inside the concrete domain constructor, the universal version of this constructor can be defined in terms of the existential one (see e.g. [113]).

in connection with concrete domains [118, 119].[27] Say, for example, that there exists a concrete feature socnum associating humans with their social security number. Then, if a human is American, this person should be *uniquely* identified by this number: no other instance of the concept name American should have the same value of the concrete feature socnum. This corresponds to a uniqueness constraint. As another example, we may want to enforce that all books having the same ISBN number share the same title. This is a functional dependency, i.e. the value of the ISBN number determines the title in a functional way. It is not a uniqueness constraint since different books may have the same ISBN number (say, two different copies of the "Handbook of Modal Logic").

In the following, we concentrate on uniqueness constraints. A *key box* is a finite set of *uniqueness constraints* ($u_1, \dots, u_n$ keyfor C), where $u_1, \dots, u_n$ are concrete paths and C is a concept description. An interpretation $\mathcal{I}$ *satisfies* ($u_1, \dots, u_n$ keyfor C) if, for all $d, e \in C^{\mathcal{I}}$,

$$u_i^{\mathcal{I}}(d) = u_i^{\mathcal{I}}(e) \text{ for } 1 \leq i \leq n \quad \text{implies} \quad d = e,$$

and it is a *model* of a key box $\mathcal{K}$ if it satisfies all uniqueness constraints in $\mathcal{K}$. In the presence of key boxes, we are interested in the satisfiability of a concept description w.r.t. a TBox and a key box, i.e. in joint models of all three components (and similarly for subsumption).

It is interesting to note that there is a close relationship between nominals and key boxes. For example, if used together with the uniqueness constraint (g keyfor $\top$), the $\mathcal{ALC}(\mathsf{Q})$ concept description $\exists g.{=_q}$ behaves similar to a nominal for each $q \in \mathbb{Q}$: it is interpreted by a set of cardinality at most one. Key boxes are strong enough to render reasoning in $\mathcal{ALC}(\mathcal{D})$ undecidable [118]: satisfiability of $\mathcal{ALC}(\mathcal{D})$ concept descriptions w.r.t. key boxes is undecidable (even without TBoxes) if the concrete domain $\mathcal{D}$ satisfies $\mathbb{N} \subseteq \Delta^{\mathcal{D}}$ and $\Phi^{\mathcal{D}}$ provides a unary predicate for equality with 0, a binary equality predicate, and a binary predicate for incrementation. Note that this result is similar to the undecidability of satisfiability in $\mathcal{ALC}(\mathcal{D})$ w.r.t. GCIs.

Decidability can be regained by allowing only Boolean combinations of concept names inside uniqueness constraints. Key boxes satisfying this property are called *Boolean*. Even w.r.t. Boolean key boxes, reasoning is much harder than reasoning without key boxes: there exists a concrete domain $\mathcal{D}$ such that $\mathcal{D}$-satisfiability is in PTime and satisfiability of $\mathcal{ALC}(\mathcal{D})$ concept descriptions w.r.t. Boolean key boxes is NExpTime-hard [118]. This high complexity cannot even be reduced if paths are restricted to length one inside $\mathcal{ALC}(\mathcal{D})$ concept descriptions and key boxes (*path-freeness*). The matching upper bound relies on a modified notion of admissibility, called *key-admissibility*. Roughly spoken, a concrete domain is key-admissible if it is admissible and provides for a binary equality predicate. The original definition given in [118] is slightly more general, but too complex to be repeated here. In [118] it is shown that satisfiability of $\mathcal{ALC}(\mathcal{D})$ concept descriptions w.r.t. Boolean key boxes is in NExpTime if $\mathcal{D}$ is key-admissible and $\mathcal{D}$-satisfiability is in NP. In the same work, also a more powerful DL with concrete domains, $\mathcal{SHOQ}(\mathcal{D})$, is extended with key boxes, and a decidability result for the path-free case is established (where $\mathcal{D}$ is required to be key-admissible, but key boxes are not expected to be Boolean).

In the case of functional dependencies, quite similar results can be established. We refer to [119] for more details.

[27] They can also be used in description logics without concrete domains, c.f. [41, 49, 103, 151].

Aggregation

Aggregation is a useful mechanism available in many expressive conceptual modelling formalisms such as database schema languages and query languages. The use of aggregation in the context of concrete domains has been proposed in [31]. As an example, consider the following description of a process and its subprocesses:

$$\mathsf{Process} \sqcap >_0(\mathsf{duration}) \sqcap \forall \mathsf{subproc}.(\mathsf{Process} \sqcap >_0(\mathsf{duration})).$$

The aggregation function "sum" is needed if we want to express that the duration of the mother process is identical to the sum of the durations of its subprocesses (of which there may be arbitrarily many).

A *concrete domain with aggregation* is a concrete domain that, additionally, provides for a set of aggregation functions $\mathsf{agg}(\mathcal{D})$, where each $\Gamma \in \mathsf{agg}(\mathcal{D})$ is associated with a partial function $\Gamma^{\mathcal{D}}$ from the set of finite multisets over $\Delta^{\mathcal{D}}$ into $\Delta^{\mathcal{D}}$. To distinguish concrete domains with aggregation from those without, we denote the former with Σ. Typical aggregation functions are min, max, sum, count, and $\mathsf{average}$. The set of $\mathcal{ALC}(\Sigma)$ concept descriptions is now defined in the same way as $\mathcal{ALC}(\mathcal{D})$ concept descriptions, except that *aggregated features* may be be used in place of concrete features, where an aggregated feature is an expression $\Gamma(r \circ g)$ with r a role, g a concrete feature, and Γ an aggregation function from Σ. The semantics of aggregated features is defined via multisets: for each interpretation $\mathcal{I}$ and each $d \in \Delta^{\mathcal{I}}$ such that the set $\{e \mid (d,e) \in r^{\mathcal{I}}\}$ is finite, we use $M_d^{r \circ g}$ to denote the multiset that, for each $z \in \Delta^{\mathcal{D}}$, contains z exactly $|\{e \mid (d,e) \in r^{\mathcal{I}} \text{ and } g^{\mathcal{I}}(e) = z\}|$ times. The semantics of aggregated features is now defined as follows:

$$(\Gamma(r \circ g))^{\mathcal{I}}(d) := \begin{cases} \Gamma^{\Sigma}(M_d^{r \circ g}) & \text{if } \{e \mid (d,e) \in r^{\mathcal{I}}\} \text{ is finite} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Returning to the initial example, we can now express the fact that the duration of the mother process is identical to the sum of the durations of all its subprocesses by writing $=(\mathsf{duration}, \mathsf{sum}(\mathsf{subproc} \circ \mathsf{duration}))$. The investigations performed by Baader and Sattler [31] reveal that the expressive power provided by aggregation functions is hard to tame in order to obtain a decidable formalism: for concrete domains with aggregation Σ where (i) $\mathbb{N} \subseteq \Delta^{\Sigma}$, (ii) Φ^{Σ} contains a (unary) predicate for equality with 1 and a (binary) equality predicate, and (iii) $\mathsf{agg}(\Sigma)$ contains min, max, and sum, satisfiability of $\mathcal{ALC}(\Sigma)$ concept descriptions is undecidable. This lower bound applies even if we admit only conjunction, the $\forall r.C$ constructor, and the concrete domain constructor, but drop all other concept constructors. Rather strong measures have to be taken to regain decidability: either, we have to drop the $\forall r.C$ constructor from the language, thus obtaining a sub-Boolean DL, or we have to confine ourselves to "well-behaved" aggregation functions such as min and max of which there exist only very few. More details can be found in [31].

6.2 Role Value Maps

Role value maps are a family of concept constructors that were available in the first description logic system, KL-ONE [45], and have since then been considered in several variations. The original and most powerful variant of role value maps has later been

found to cause undecidability, even if used with quite weak (sub-Boolean) description logics such as the one available in the KL-ONE system.

To define role value maps, we must introduce the notion of a *path*, i.e., a composition $r_1 \circ \cdots \circ r_n$ of role names. If R and S are paths, then the expression $(R \subseteq S)$ is a *containment role value map* and $(R = S)$ is an *equality role value map*. To extend $\mathcal{ALC}$ with role value maps, we admit them as additional concept constructors. The resulting description logic is denoted $\mathcal{ALC}^{\mathsf{rvm}}$. The semantics of the additional concept constructors is as follows:

$$\begin{array}{lcl} (R \subseteq S)^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid \forall e.(d,e) \in R^{\mathcal{I}} \text{ implies } (d,e) \in S^{\mathcal{I}}\}, \\ (R = S)^{\mathcal{I}} & = & \{e \in \Delta^{\mathcal{I}} \mid \forall e.(d,e) \in R^{\mathcal{I}} \text{ iff } (d,e) \in S^{\mathcal{I}}\}, \end{array}$$

where the interpretation $R^{\mathcal{I}}$ of paths R is defined in the obvious way through relational composition. For example, the concept description $\mathsf{Person} \sqcap (\mathsf{child} \circ \mathsf{friend} \subseteq \mathsf{knows})$ describes persons knowing all the friends of their children.

Though there appears to be no direct modal counterpart to role value maps as a concept constructor, there is a connection to *modal reduction principles* as discussed in Chapter 7. Modal reduction principles (MRPs) are axioms of the form $Mp \rightarrow Np$, where M and N are sequences of modal operators $\Box_i$ and $\Diamond_j$ [152]. We call an MRP *box-only* if M and N are non-empty sequences of box operators. There is a close correspondence between normal modal logics axiomatized by box-only MRPs and $\mathcal{ALC}^{\mathsf{rvm}}$: it follows from Sahlqvist's completeness theorem that normal modal logics axiomatized by a box-only MRP $\varphi = \Box_{i_1} \cdots \Box_{i_n} p \rightarrow \Box_{j_1} \cdots \Box j_m p$ are characterized by the class of frames that validates φ; moreover, it is a routine task to show that the same class of frames is determined by all models of the GCI $\top \sqsubseteq (r_{i_1} \circ \cdots \circ r_{i_n} \subseteq s_{j_1} \circ \cdots \circ s_{j_m})$. There is also a close connection between role value maps and so-called grammar logics [58], which will be discussed in more detail below.

Undecidability

Reasoning in the first description logic system KL-ONE was initially believed to be in PTIME. However, in 1989 Schmidt-Schauß was able to show that it is undecidable, identifying role value maps as the main culprit [141]. More precisely, Schmidt-Schauß proves that, even in the description logic $\mathcal{FL}_0^{\mathsf{rvm}}$ providing only for the constructors conjunction, value restriction, and role value maps, subsumption w.r.t. the empty TBox is undecidable.

The proof of Schmidt-Schauß uses a reduction of the word problem for groups. We present here a slight variation that reduces the word problem for semigroups [36].[28] For simplicity, we first show undecidability of $\mathcal{FL}_0^{\mathsf{rvm}}$ w.r.t. GCIs, and then eliminate the GCIs from the reduction. A finitely presented semigroup S is given in the form of defining identities $s_1 = t_1, \ldots, s_m = t_m$, where the s_i and t_i are words over some finite alphabet Σ. Then the word problem is to decide, given S and words s and t, whether $s = t$ holds in S, i.e., whether the identity $s = t$ can be derived from the defining identities of S and the usual axioms for semigroups. For the reduction, we view the symbols $r_1, \ldots, r_n$ of Σ

[28] The reduction of Schmidt-Schauß yields a slightly stronger result since it applies also to the case where we have only equality role value maps, but no containment role value maps.

as role names and construct a set of GCIs $\mathcal{T}_S$ and a concept description $D_{s,t}$ as follows:

$$\begin{array}{rcl} \mathcal{T}_S & := & \{\top \sqsubseteq (s_1 = t_1) \sqcap \cdots \sqcap (s_m = t_m)\} \\ D_{s,t} & := & (s = t) \end{array}$$

Then it is not hard to show that $\top \sqsubseteq_{\mathcal{T}_S} D_{s,t}$ if and only if $s = t$ can be derived from the defining identities of S. This yields undecidability of subsumption in $\mathcal{FL}_0^{\mathsf{rvm}}$ w.r.t. GCIs. To get rid of GCIs, we can internalize them (c.f. Section 3.1). More precisely, the subsumption $\top \sqsubseteq_{\mathcal{T}_S} D_{s,t}$ holds if and only if $C_S \sqsubseteq D_{s,t}$, where C_S is defines as follows:

$$C_S \quad := \quad \bigsqcap_{r \in \Sigma} \big((r \subseteq u) \sqcap (u \circ u \subseteq u)\big) \sqcap \forall u.((s_1 = t_1) \sqcap \cdots \sqcap (s_m = t_m))$$

where u is a role name that does not occur in Σ. It follows that subsumption in $\mathcal{FL}_0^{\mathsf{rvm}}$ is undecidable also without GCIs.

There is a related result in modal logic that should be mentioned: Shehtman proved in 1982 that there exists a set Γ of box-only modal reduction principles such that, in the normal modal logic axiomatized by Γ, satisfiability is undecidable [146]. By what was said above about the connection between MRPs and role value maps, and since GCIs can be internalized in $\mathcal{ALC}^{\mathsf{rvm}}$ similar to what was done above in the case of $\mathcal{FL}_0^{\mathsf{rvm}}$, it is obvious that this gives a proof of undecidability of reasoning in $\mathcal{ALC}^{\mathsf{rvm}}$ without TBoxes and GCIs.

To avoid undecidability, two approaches have been considered: first, role value maps have been weakened into feature agreements and feature disagreements, which have a similar semantics but are restricted to paths comprised only of functional roles; and second, the original role value maps have been used for paths of a syntactically restricted form. In the next section, we describe the first approach. Syntactic restrictions on paths are discussed subsequently in Section 6.2.

Feature Agreements

A *feature path* is a composition $f_1 \circ \cdots \circ f_n$ of abstract features as introduced in Section 6.1. If u and v are feature paths, then the expression $(u = v)$ is a *feature agreement*, and $(u \neq v)$ is a *feature disagreement*. The description logic $\mathcal{ALCF}$ is obtained from $\mathcal{ALC}^{\mathsf{rvm}}$ by replacing role value maps with feature (dis)agreements. The semantics of the new concept constructors is:

$$\begin{array}{rcl} (u = v)^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid \exists e.(d,e) \in u^{\mathcal{I}} \text{ and } (d,e) \in v^{\mathcal{I}}\} \\ (u \neq v)^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid \exists e, e'.(d,e) \in u^{\mathcal{I}}, (d,e') \in v^{\mathcal{I}}, \text{ and } e \neq e'\} \end{array}$$

In the literature, feature agreements are sometimes called the *same-as* constructor, e.g. in their incarnation in the Classic system [44]. The restriction of role value maps to feature paths has an impairing effect on the usefulness of feature (dis)agreements. For example, the concept description

$$\mathsf{Person} \sqcap (\mathsf{child} \circ \mathsf{friend} \subseteq \mathsf{knows})$$

cannot be expressed in $\mathcal{ALCF}$ since child and knows should not be forced to be functional. However, feature (dis)agreements can still be usefully employed, as illustrated by the following concept definition:

$$\mathsf{ParentsMarried} \equiv (\mathsf{mother} \circ \mathsf{married\text{-}to} = \mathsf{father})$$

The main advantage of feature agreements over role value maps lies in their computational properties. Indeed, Hollunder and Nutt [89] show that satisfiability (and thus also subsumption) of $\mathcal{ALCF}$ concept descriptions is decidable and PSPACE-complete (see also [113]).

When further expressive means are added to $\mathcal{ALCF}$, the computational complexity often gets dramatically worse. In this respect, feature (dis)agreements resemble concrete domains: the PSPACE upper bound of the basic description logic with feature (dis)agreements $\mathcal{ALCF}$ is rather fragile w.r.t. extensions of the language. An important example for this behaviour are TBoxes. As shown in [110], satisfiability in $\mathcal{ALCF}$ w.r.t. acyclic TBoxes is NEXPTIME-complete. The result is established by reduction of a NEXPTIME-complete variant of the domino problem, and exploits the succinctness gained by introducing acyclic TBoxes similar to what is discussed in Section 6.1 in the context of concrete domains. When cyclic TBoxes or GCIs are admitted, satisfiability and subsumption in $\mathcal{ALCF}$ even become undecidable [11], which can be shown by a reduction of the word problem for groups. Other extensions of $\mathcal{ALCF}$ that make reasoning harder include intersection of roles, inverse roles, and transitive closure of functional roles. In the first case, satisfiability of concept descriptions becomes NEXPTIME-complete, while it is undecidable in the latter two cases [11, 112].

Restricted Paths in Role Value Maps

In this section, we consider syntactic restrictions on paths inside role value maps. There are some quite drastic restrictions that are easily seen to regain decidability of reasoning in $\mathcal{ALC}^{\mathsf{rvm}}$:

- Only allow paths of length one. In this case, reasoning in $\mathcal{ALC}^{\mathsf{rvm}}$ can be reduced to reasoning in $\mathcal{ALC}^{\neg,\cap,\cup}$, the extension of $\mathcal{ALC}$ with Boolean role constructors: simply replace $(r = s)$ with $(r \subseteq s) \sqcap (s \subseteq r)$, and $(r \subseteq s)$ with $\forall(r \cap \neg s).\bot$. Since satisfiability and subsumption in $\mathcal{ALC}^{\neg,\cap,\cup}$ w.r.t. GCIs is known to be decidable [122, 121], so is the restricted version of $\mathcal{ALC}^{\mathsf{rvm}}$. Note that there is also a close connection to role hierarchies: a role inclusion $r \sqsubseteq s$ as used in a role hierarchy can be simulated using the concept equation $\top \equiv (r \subseteq s)$ in $\mathcal{ALC}^{\mathsf{rvm}}$.

- Only admit role value maps of the form $(r \circ r \subseteq r)$. Clearly, we obtain a localized variant of transitive roles. It is straightforward to adapt the standard techniques for dealing with (globally) transitive roles [135, 94] to show that, in this variant of $\mathcal{ALC}^{\mathsf{rvm}}$, satisfiability and subsumption w.r.t. GCIs are decidable. It appears to be an open problem whether admitting single-role role value maps of the form $(r^n \subseteq r^m)$, with $n, m \in \mathbb{N}$ and r^n denoting the n-fold composition of r, yields a decidable variant of $\mathcal{ALC}^{\mathsf{rvm}}$.

More powerful decidable fragments of $\mathcal{ALC}^{\mathsf{rvm}}$ can be obtained by restricting paths in a less strict way. This is done by Horrocks and Sattler [96] in their work on complex role inclusion axioms (RIAs), and in the closely related area of grammar logics [58, 34, 59]. In both cases, role value maps are *not* considered to be concept constructors, but rather they are *global*, similar to the role inclusions in a role hierarchy, i.e., the role value map must hold for every element of the interpretation domain. In the following, we consider $\mathcal{ALC}$ concept descriptions and assume the presence of a *role box*, i.e. a finite set of role value maps $(R \subseteq S)$. An interpretation $\mathcal{I}$ is a *model* of a role box $\mathcal{R}$ if it satisfies $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$

for all $(R \subseteq S) \in \mathcal{R}$. In this setting, we are interested in satisfiability and subsumption w.r.t. TBoxes/GCIs and role boxes, i.e., in common models of all three inputs.

Translated to this terminology, the idea of grammar logic is to view a role value map $(r_1 \circ \cdots \circ r_k \subseteq s_1 \circ \cdots \circ s_\ell)$ as a production rule $s_1 \cdots s_\ell \to r_1 \cdots r_k$, and a role box as a formal grammar [58]. Here, the mapping from role names to terminal and non-terminal symbols is arbitrary, but fixed. From the description logic perspective, the most relevant results from grammar logic are the following. First, Demri shows that satisfiability and subsumption of $\mathcal{ALC}$ concept descriptions is ExpTime-complete if only role boxes corresponding to left-linear or right-linear grammars are admitted [59]. In role boxes of this form, we can express properties such as "the enemies of my friends are my enemies":

$$(\mathsf{friend} \circ \mathsf{enemy} \subseteq \mathsf{enemy}).$$

Note that this result captures the case where paths are required to be of length at most one, but not the case $(r \circ r \subseteq r)$. Second, Baldoni et al. show that satisfiability of $\mathcal{ALC}$ concept descriptions is undecidable if role boxes corresponding to context-free grammars are admitted [34]. This result was later strengthened by Demri to linear grammars [59].

Horrocks and Sattler consider the extension of $\mathcal{SHIQ}$ with global role value maps of the form $(r \subseteq s)$, $(r \circ s \subseteq r)$, and $(s \circ r \subseteq r)$, where r, s is a role name or the inverse of a role name [96]. For example, the statement about enemies of friends from above can be strengthened by additionally saying that the friends of my enemies are my enemies:

$$\begin{array}{lcl}(\mathsf{friend} \circ \mathsf{enemy} & \subseteq & \mathsf{enemy}) \\ (\mathsf{enemy} \circ \mathsf{friend} & \subseteq & \mathsf{enemy}).\end{array}$$

Note that this role box does not correspond to a left-linear or right-linear grammar. Role value maps of this form are of particular interest since they allow to describe the propagation of properties, e.g. along part-whole relations: "the owner of a whole is the owner of all parts" can be written as $(\mathsf{part\text{-}of} \circ \mathsf{owner} \subseteq \mathsf{owner})$. Let us call Horrocks and Sattler's variant of role value maps *HS-RVMs*. Horrocks and Sattler obtain the following results: first, reasoning in the extension of $\mathcal{SHIQ}$ with HS-RVMs is undecidable in the general case. An inspection of the proof shows that undecidability already arises in $\mathcal{ALC}$ extended with inverse roles, number restrictions of the form $(\leqslant 1\ r)$, GCIs, and HS-RVMs. Decidability of plain $\mathcal{ALC}$ extended with HS-RVMs (and possibly GCIs) appears to be an open problem. Second, satisfiability and subsumption in $\mathcal{SHIQ}$ w.r.t. GCIs and role boxes becomes decidable if we admit only HS-RVMs and *acyclic* role boxes. Acyclicity of role boxes is defined similar to the TBox case: a role r *directly affects* a role s if $r \neq s$ and there is an HS-RVM with (i) r appearing on the left-hand side (possibly inside a composition) and s appearing on the right-hand side, or (ii) the inverse of r appearing on the left-hand side and the inverse of s appearing on the right-hand side. *Affects* is the transitive closure of "directly affects". Then a role box is acyclic if no role affects itself. Observe that the above example about friends and enemies is acyclic.

There are several other restrictions of paths that can be considered. An interesting example is to admit only role value maps $(R \subseteq S)$ with R and S paths of equal length. This restriction has been investigated by Molitor [126], but the decidability status of $\mathcal{ALC}^{\mathsf{rvm}}$ under this restriction is, as of now, unknown.

ACKNOWLEDGEMENTS

We are grateful to Silvio Ghilardi and Andrey Bovykin for useful remarks on earlier versions of this article.

BIBLIOGRAPHY

[1] Carlos Areces, Patrick Blackburn, and Maarten Marx. A road-map on complexity for hybrid logics. In *Proc. of the Annual Conf. of the Eur. Assoc. for Computer Science Logic (CSL'99)*, volume 1683 of *Lecture Notes in Computer Science*, pages 307–321. Springer-Verlag, 1999.

[2] Franz Baader. Unification in commutative theories. *J. of Symbolic Computation*, 8:479–497, 1989.

[3] Franz Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)*, 1991.

[4] Franz Baader. A formal definition for the expressive power of terminological knowledge representation languages. *J. of Logic and Computation*, 6:33–54, 1996.

[5] Franz Baader. Using automata theory for characterizing the semantics of terminological cycles. *Ann. of Mathematics and Artificial Intelligence*, 18:175–219, 1996.

[6] Franz Baader. Computing the least common subsumer in the description logic $\mathcal{EL}$ w.r.t. terminological cycles with descriptive semantics. In *Proceedings of the 11th International Conference on Conceptual Structures, ICCS 2003*, volume 2746 of *Lecture Notes in Artificial Intelligence*, pages 117–130. Springer-Verlag, 2003.

[7] Franz Baader. Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In Georg Gottlob and Toby Walsh, editors, *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 319–324. Morgan Kaufmann, 2003.

[8] Franz Baader. Restricted role-value-maps in a description logic with existential restrictions and terminological cycles. In *Proceedings of the 2003 International Workshop on Description Logics (DL2003)*, CEUR-WS, 2003.

[9] Franz Baader. Terminological cycles in a description logic with existential restrictions. In Georg Gottlob and Toby Walsh, editors, *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 325–330. Morgan Kaufmann, 2003.

[10] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*. Morgan Kaufmann, Los Altos, 2005.

[11] Franz Baader, Hans-Jürgen Bürckert, Bernhard Nebel, Werner Nutt, and Gert Smolka. On the expressivity of feature logics with negation, functional uncertainty, and sort equations. *J. of Logic, Language and Information*, 2:1–18, 1993.

[12] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[13] Franz Baader, Enrico Franconi, Bernhard Hollunder, Bernhard Nebel, and Hans-Jürgen Profitlich. An empirical analysis of optimization techniques for terminological representation systems or: Making KRIS get a move on. *Applied Artificial Intelligence. Special Issue on Knowledge Base Management*, 4:109–132, 1994.

[14] Franz Baader and Philipp Hanschke. A schema for integrating concrete domains into concept languages. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)*, pages 452–457, 1991.

[15] Franz Baader and Philipp Hanschke. A scheme for integrating concrete domains into concept languages. Technical Report RR-91-10, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), 1991.

[16] Franz Baader and Philipp Hanschke. Extensions of concept languages for a mechanical engineering application. In *Proc. of the 16th German Workshop on Artificial Intelligence (GWAI'92)*, volume 671 of *Lecture Notes in Computer Science*, pages 132–143, Bonn (Germany), 1992. Springer-Verlag.

[17] Franz Baader and Bernhard Hollunder. $\mathcal{KRIS}$: $\mathcal{K}$nowledge $\mathcal{R}$epresentation and $\mathcal{I}$nference $\mathcal{S}$ystem. *SIGART Bull.*, 2(3):8–14, 1991.

[18] Franz Baader and Bernhard Hollunder. A terminological knowledge representation system with complete inference algorithm. In *Proc. of the Workshop on Processing Declarative Knowledge (PDK'91)*, volume 567 of *Lecture Notes in Artificial Intelligence*, pages 67–86. Springer-Verlag, 1991.

[19] Franz Baader, Ian Horrocks, and Ulrike Sattler. Description logics as ontology languages for the semantic web. In Dieter Hutter and Werner Stephan, editors, *Festschrift in honor of Jörg Siekmann*, Lecture Notes in Artificial Intelligence. Springer-Verlag, 2003.

[20] Franz Baader and Ralf Küsters. Computing the least common subsumer and the most specific concept in the presence of cyclic $\mathcal{ALN}$-concept descriptions. In *Proc. of the 22nd German Annual Conf. on Artificial Intelligence (KI'98)*, volume 1504 of *Lecture Notes in Computer Science*, pages 129–140. Springer-Verlag, 1998.

[21] Franz Baader and Ralf Küsters. Matching in description logics with existential restrictions. In *Proc. of the 7th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2000)*, pages 261–272, 2000.

[22] Franz Baader and Ralf Küsters. Unification in a description logic with transitive closure of roles. In Robert Nieuwenhuis and Andrei Voronkov, editors, *Proceedings of the 8th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, (LPAR 2001)*, Lecture Notes in Artificial Intelligence, Havana, Cuba, 2001. Springer-Verlag.

[23] Franz Baader, Ralf Küsters, Alex Borgida, and Deborah L. McGuinness. Matching in description logics. *J. of Logic and Computation*, 9(3):411–447, 1999.

[24] Franz Baader, Ralf Küsters, and Ralf Molitor. Structural subsumption considered from an automata theoretic point of view. In *Proc. of the 1998 Description Logic Workshop (DL'98)*. CEUR Electronic Workshop Proceedings, http://ceur-ws.org/Vol-11/, 1998.

[25] Franz Baader, Ralf Küsters, and Ralf Molitor. Computing least common subsumers in description logics with existential restrictions. In *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI'99)*, pages 96–101, 1999.

[26] Franz Baader, Ralf Küsters, and Ralf Molitor. Rewriting concepts using terminologies. In *Proc. of the 7th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2000)*, pages 297–308, 2000.

[27] Franz Baader, Carsten Lutz, Holger Sturm, and Frank Wolter. Fusions of description logics and abstract description systems. *Journal of Artificial Intelligence Research (JAIR)*, 16:1–58, 2002.

[28] Franz Baader and Paliath Narendran. Unification of concepts terms in description logics. *J. of Symbolic Computation*, 31(3):277–305, 2001.

[29] Franz Baader and Werner Nutt. Basic description logics. In *[12]*, pages 43–95. 2003.

[30] Franz Baader and Ulrike Sattler. An overview of tableau algorithms for description logics. *Studia Logica*, 69:5–40, 2001.

[31] Franz Baader and Ulrike Sattler. Description logics with aggregates and concrete domains. *Information Systems*, 28(8):979–1004, 2003.

[32] Franz Baader and Wayne Snyder. Unification theory. In J.A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning, Volume I*, pages 447–533. Elsevier Science Publishers, 2001.

[33] Philippe Balbiani and Jean-Francois Condotta. Computational complexity of propositional linear temporal logics based on qualitative spatial or temporal reasoning. In *Proc. of the 4th Int. Workshop on Frontiers of Combining Systems (FroCoS 2002)*, number 2309 in Lecture Notes in Artificial Intelligence, pages 162–176. Springer-Verlag, 2002.

[34] Matteo Baldoni, Laura Giordano, and Alberto Martelli. A tableau calculus for multimodal logics and some (un)decidability results. In *Proc. of the Int. Conf. on Analytic Tableaux and Related Methods (TABLEAUX'98)*, volume 1397 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1998.

[35] Piero A. Bonatti. On the undecidability of description and dynamic logics with recursion and counting. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 331–336. Morgan Kaufmann, Los Altos, 2003.

[36] William W. Boone. The word problem. *Ann. of Mathematics*, 2(70):207–265, 1959.

[37] Alexander Borgida. On the relationship between description logics and predicate logic queries. Technical Report LCSR-TR-295-A, Rutgers University, New Brunswick (NJ, USA), 1992.

[38] Alexander Borgida. On the relative expressiveness of description logics and predicate logics. *Artificial Intelligence*, 82(1–2):353–367, 1996.

[39] Alexander Borgida and Deborah L. McGuinness. Asking queries about frames. In *Proc. of the 5th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'96)*, pages 340–349, 1996.

[40] Alexander Borgida and Peter F. Patel-Schneider. A semantics and complete algorithm for subsumption in the CLASSIC description logic. *J. of Artificial Intelligence Research*, 1:277–308, 1994.

[41] Alexander Borgida and Grant E. Weddell. Adding uniqueness constraints to description logics (preliminary report). In *Proc. of the 5th Int. Conf. on Deductive and Object-Oriented Databases (DOOD'97)*, pages 85–102, 1997.

[42] Ronald J. Brachman. Structured inheritance networks. In W. A. Woods and R. J. Brachman, editors, *Research in Natural Language Understanding*, Quarterly Progress Report No. 1, BBN Report No. 3742, pages 36–78. Bolt, Beranek and Newman Inc., Cambridge, Mass., 1978.

[43] Ronald J. Brachman and Hector J. Levesque, eds. *Readings in Knowledge Representation*. Morgan Kaufmann, 1985.

[44] Ronald J. Brachman, Deborah L. McGuinness, Peter F. Patel-Schneider, Lori Alperin Resnick, and Alexander Borgida. Living with CLASSIC: When and how to use a KL-ONE-like language. In John F. Sowa, editor, *Principles of Semantic Networks*, pages 401–456. Morgan Kaufmann, Los Altos, 1991.

[45] Ronald J. Brachman and James G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[46] Sebastian Brandt. Polynomial time reasoning in a description logic with existential restrictions, gci axioms, and—what else? In *Proc. of the 16th Eur. Conf. on Artificial Intelligence (ECAI 2004)*, 2004.

[47] Sebastian Brandt, Ralf Küsters, and Anni-Yasmin Turhan. Approximating $\mathcal{ALCN}$-concept descriptions. In *Proc. of the 2002 Description Logic Workshop (DL 2002)*, 2002.

[48] Sebastian Brandt, Ralf Küsters, and Anni-Yasmin Turhan. Approximation and difference in description logics. In D. Fensel, F. Giunchiglia, D. McGuiness, and M.-A. Williams, editors, *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR2002)*, pages 203–214, San Francisco, CA, 2002. Morgan Kaufman.

[49] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Identification constraints and functional dependencies in description logics. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001)*, pages 155–160, 2001.

[50] Diego Calvanese, Giuseppe de Giacomo, and Maurizio Lenzerini. 2-ATAs make DLs easy. In *Proc. of the 2002 Description Logic Workshop (DL 2002)*, 2002.

[51] Diego Calvanese, Giuseppe de Giacomo, Maurizio Lenzerini, Riccardo Rosati, and Guido Vetere. DL-Lite: Practical reasoning in rich DLs. In Ralf Möller and Volker Haarslev, editors, *Proceedings of the International Workshop in Description Logics 2004 (DL2004)*, volume 104 of *CEUR-WS (http://ceur-ws.org/)*, 2004.

[52] Diego Calvanese, Maurizio Lenzerini, and Daniele Nardi. Description logics for conceptual data modeling. In Jan Chomicki and Günter Saake, editors, *Logics for Databases and Information Systems*, pages 229–264. Kluwer Academic Publisher, 1998.

[53] William W. Cohen and Haym Hirsh. Learning the CLASSIC description logics: Theoretical and experimental results. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proc. of the 4th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'94)*, pages 121–133, 1994.

[54] Giuseppe De Giacomo and Maurizio Lenzerini. Boosting the correspondence between description logics and propositional dynamic logics. In *Proc. of the 12th Nat. Conf. on Artificial Intelligence (AAAI'94)*, pages 205–212. AAAI Press/The MIT Press, 1994.

[55] Giuseppe De Giacomo and Maurizio Lenzerini. Concept language with number restrictions and fixpoints, and its relationship with μ-calculus. In *Proc. of the 11th Eur. Conf. on Artificial Intelligence (ECAI'94)*, pages 411–415, 1994.

[56] Giuseppe De Giacomo and Maurizio Lenzerini. What's in an aggregate: Foundations for description logics with tuples and sets. In *Proc. of the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI'95)*, pages 801–807, 1995.

[57] Giuseppe De Giacomo and Maurizio Lenzerini. TBox and ABox reasoning in expressive description logics. In Luigia C. Aiello, John Doyle, and Stuart C. Shapiro, editors, *Proc. of the 5th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'96)*, pages 316–327. Morgan Kaufmann, Los Altos, 1996.

[58] Luis Farinas del Cerro and Martti Penttonen. Grammar logics. *Logique et Analyse*, 121-122:123–134, 1988.

[59] Stephane Demri. The complexity of regularity in grammar logics and related modal logics. *J. of Logic and Computation*, 11(6), 2001.

[60] Stephane Demri. LTL over integer periodicity constraints. In I. Walukiewicz, editor, *Proceedings of the 7th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'04)*, number 2987 in Lecture Notes in Computer Science, pages 121–135, Barcelona, Spain, 2004. Springer-Verlag.

[61] Francesco Donini. Complexity of reasoning. In *[12]*, pages 96–136. 2003.

[62] Francesco M. Donini, Bernhard Hollunder, Maurizio Lenzerini, Alberto Marchetti Spaccamela, Daniele Nardi, and Werner Nutt. The complexity of existential quantification in concept languages. *Artificial Intelligence*, 2–3:309–327, 1992.

[63] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. The complexity of concept languages. In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proc. of the 2nd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'91)*, pages 151–162. Morgan Kaufmann, Los Altos, 1991.

[64] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. Tractable concept languages. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)*, pages 458–463, Sydney (Australia), 1991.

[65] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. The complexity of concept languages. *Information and Computation*, 134:1–58, 1997.

[66] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. Tractability and intractability in description logics. Available at ftp://ftp.dis.uniroma1.it/pub/ai/papers/dlnn99d.ps.gz, 1999.

[67] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. Deduction in concept languages: From subsumption to instance checking. *J. of Logic and Computation*, 4(4):423–452, 1994.

[68] Jürgen Edelmann and Bernd Owsnicki-Klewe. Data models in knowledge representation systems: A case study. In C. R. Rollinger and W. Horn, editors, *GWAI-86 and 2. Österreichische Artificial-Intelligence-Tagung*, volume 124 of *Informatik-Fachberichte*, pages 69–74. Springer-Verlag, 1986.

[69] Ulrich Faigle and György Turán. Sorting and recognition problems for ordered sets. *SIAM J. on Computing*, 17(1), 1988.

[70] Maurizio Fattorosi-Barnaba and F. de Caro. Graded modalities I. *Studia Logica*, 44(2):197–221, 1985.

[71] Kit Fine. In so many possible worlds. *Notre Dame J. of Formal Logic*, 13(4):516–520, 1972.

[72] Dov Gabbay and Larisa Maksimova. *Interpolation and Definability: Modal and Intuitionistic Logic*. Oxford University Press, 2005.

[73] Michael R. Garey and David S. Johnson. *Computers and Intractability — A guide to NP-completeness*. W. H. Freeman and Company, San Francisco (CA, USA), 1979.

[74] George Gargov and Valentin Goranko. Modal logic with names. *J. of Philosophical Logic*, 22:607–636, 1993.

[75] Silvio Ghilardi. Unification in intuitionistic logic. *J. of Symbolic Logic*, 64:859–880, 1999.

[76] Silvio Ghilardi. Best solving modal equations. *Ann. Pure Appl. Logic*, 102(3):183–198, 2000.

[77] Volker Haarslev and Ralf Möller. Expressive ABox reasoning with number restrictions, role hierarchies, and transitively closed roles. In *Proc. of the 7th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2000)*. Morgan Kaufmann, Los Altos, 2000.

[78] Volker Haarslev and Ralf Möller. High performance reasoning with very large knowledge bases: A practical case study. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001)*, 2001.

[79] Volker Haarslev and Ralf Möller. RACER system description. In Rajeev Goré, Alexander Leitsch, and Tobias Nipkow, editors, *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2001)*, volume 2083 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2001.

[80] Volker Haarslev, Ralf Möller, and Michael Wessel. The description logic $\mathcal{ALCNH}_{R^+}$ extended with concrete domains: A practically motivated approach. In Rajeev Goré, Alexander Leitsch, and Tobias Nipkow, editors, *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2001)*, volume 2083 of *Lecture Notes in Artificial Intelligence*, pages 29–44. Springer-Verlag, 2001.

[81] Joseph Y. Halpern and Yoram Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.

[82] Philipp Hanschke. Specifying role interaction in concept languages. In *Proc. of the 3rd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'92)*, pages 318–329. Morgan Kaufmann, Los Altos, 1992.

[83] Patrick J. Hayes. The logic of frames. In D. Metzing, editor, *Frame Conceptions and Text Understanding*, pages 46–61. Walter de Gruyter and Co., 1979. Republished in [43].

[84] Edith Hemaspaandra. The complexity of poor man's logic. *J. of Logic and Computation*, 11(4):609–622, 2001.

[85] Monika R. Henzinger, Thomas A. Henzinger, and Peter W. Kopke. Computing simulations on finite and infinite graphs. In *36th Annual Symposium on Foundations of Computer Science*, pages 453–462, Milwaukee, Wisconsin, 1995. IEEE Computer Society Press.

[86] Bernhard Hollunder. Hybrid inferences in KL-ONE-based knowledge representation systems. In *Proc. of the German Workshop on Artificial Intelligence*, pages 38–47. Springer-Verlag, 1990.

[87] Bernhard Hollunder. Consistency checking reduced to satisfiability of concepts in terminological systems. *Ann. of Mathematics and Artificial Intelligence*, 18(2–4):133–157, 1996.

[88] Bernhard Hollunder and Franz Baader. Qualifying number restrictions in concept languages. In *Proc. of the 2nd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'91)*, pages 335–346, 1991.

[89] Bernhard Hollunder and Werner Nutt. Subsumption algorithms for concept languages. Technical Report RR-90-04, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Kaiserslautern (Germany), 1990.

[90] Ian Horrocks. *Optimising Tableaux Decision Procedures for Description Logics*. PhD thesis, University of Manchester, 1997.

[91] Ian Horrocks. Using an expressive description logic: FaCT or fiction? In *Proc. of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 636–647, 1998.

[92] Ian Horrocks. Implementation and optimization techniques. In *[12]*, pages 306–346. 2003.

[93] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *J. of Web Semantics*, 1(1):7–26, 2003.

[94] Ian Horrocks and Ulrike Sattler. A description logic with transitive and inverse roles and role hierarchies. *J. of Logic and Computation*, 9(3):385–410, 1999.

[95] Ian Horrocks and Ulrike Sattler. Ontology reasoning in the $\mathcal{SHOQ}$(D) description logic. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001)*. Morgan Kaufmann, Los Altos, 2001.

[96] Ian Horrocks and Ulrike Sattler. Decidability of SHIQ with complex role inclusion axioms. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 343–348. Morgan Kaufmann, Los Altos, 2003.

[97] Ian Horrocks and Ulrike Sattler. A tableaux decision procedure for $\mathcal{SHOIQ}$. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, Edinburgh (UK), 2005. Morgan Kaufmann, Los Altos.

[98] Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Practical reasoning for expressive description logics. In Harald Ganzinger, David McAllester, and Andrei Voronkov, editors, *Proc. of the 6th Int. Conf. on Logic for Programming and Automated Reasoning (LPAR'99)*, volume 1705 of *Lecture Notes in Artificial Intelligence*, pages 161–180. Springer-Verlag, 1999.

[99] Ian Horrocks and Stephan Tobies. Reasoning with axioms: Theory and practice. In A. G. Cohn, F. Giunchiglia, and B. Selman, editors, *Proc. of the 7th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2000)*. Morgan Kaufmann, Los Altos, 2000.

[100] Ullrich Hustadt, Boris Motik, and Ulrike Sattler. Reducing SHIQ-description logic to disjunctive datalog programs. In *Proc. of the 9th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2004)*. Morgan Kaufmann, Los Altos, 2004.

[101] Ullrich Hustadt and Renate A. Schmidt. Issues of decidability for description logics in the framework of resolution. In R. Caterra and G. Salzer, editors, *Automated Deduction in Classical and Non-Classical Logic*, volume 1761 of *Lecture Notes in Artificial Intelligence*, pages 191–205. Springer-Verlag, 2000.

[102] Yevgeny Kazakov and Hans de Nivelle. Subsumption of concepts in $\mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete. In *Proc. of the 2003 Description Logic Workshop (DL 2003)*. CEUR Electronic Workshop Proceedings, http://CEUR-WS.org/Vol-81/, 2003.

[103] Vitaliy L. Khizder, David Toman, and Grant E. Weddell. On decidability and complexity of description logics with uniqueness constraints. In *Proc. of the 8th Int. Conf. on Database Theory (ICDT 2001)*, 2001.

[104] Dexter Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[105] Natasha Kurtonina and Maarten de Rijke. Expressiveness of concept expressions in first-order description logics. *Artificial Intelligence*, 107(2):303–333, 1999.

[106] Ralf Küsters. Characterizing the semantics of terminological cycles in $\mathcal{ALN}$ using finite automata. In *Proc. of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 499–510, 1998.

[107] Ralf Küsters and Ralf Molitor. Approximating most specific concepts in description logics with existential restrictions. In Franz Baader, Gerd Brewka, and Thomas Eiter, editors, *Proceedings of the Joint German/Austrian Conference on Artificial Intelligence (KI 2001)*, volume 2174 of *Lecture Notes in Artificial Intelligence*, pages 33–47, Vienna, Austria, 2001. Springer-Verlag.

[108] Ralf Küsters and Ralf Molitor. Computing least common subsumers in $\mathcal{ALEN}$. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001)*, pages 219–224, 2001.

[109] Hector J. Levesque and Ron J. Brachman. Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3:78–93, 1987.

[110] Carsten Lutz. Complexity of terminological reasoning revisited. In *Proc. of the 6th Int. Conf. on Logic for Programming and Automated Reasoning (LPAR'99)*, volume 1705 of *Lecture Notes in Artificial Intelligence*, pages 181–200. Springer-Verlag, 1999.

[111] Carsten Lutz. Adding numbers to the $\mathcal{SHIQ}$ description logic—First results. In *Proc. of the 8th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2002)*, pages 191–202. Morgan Kaufmann, Los Altos, 2002.

[112] Carsten Lutz. *The Complexity of Reasoning with Concrete Domains*. PhD thesis, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 2002.

[113] Carsten Lutz. PSPACE reasoning with the description logic $\mathcal{ALCF}(\mathcal{D})$. *Logic Journal of the IGPL*, 10(5):535–568, 2002.

[114] Carsten Lutz. Reasoning about entity relationship diagrams with complex attribute dependencies. In *Proc. of the 2002 Description Logic Workshop (DL 2002)*, 2002.

[115] Carsten Lutz. Description logics with concrete domains—a survey. In Philippe Balbiani, Nobu-Yuki Suzuki, Frank Wolter, and Michael Zakharyaschev, editors, *Advances in Modal Logics Volume 4*, pages 265–296. King's College Publications, 2003.

[116] Carsten Lutz. Combining interval-based temporal reasoning with general TBoxes. *Artificial Intelligence*, 152(2):235–274, 2004.

[117] Carsten Lutz. NExpTime-complete description logics with concrete domains. *ACM Transactions on Computational Logic*, 5(4):669–705, 2004.

[118] Carsten Lutz, Carlos Areces, Ian Horrocks, and Ulrike Sattler. Keys, nominals, and concrete domains. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 349–354. Morgan Kaufmann, Los Altos, 2003.

[119] Carsten Lutz and Maja Milicic. Description logics with concrete domains and functional dependencies. In *Proc. of the 16th Eur. Conf. on Artificial Intelligence (ECAI 2004)*, 2004.

[120] Carsten Lutz and Maja Milicic. A tableau algorithm for description logics with concrete domains and GCIs. In *Proc. of the Int. Conf. on Analytic Tableaux and Related Methods (TABLEAUX 2005)*, number 3702 in Lecture Notes in Artificial Intelligence. Springer-Verlag, 2005.

[121] Carsten Lutz and Ulrike Sattler. The complexity of reasoning with Boolean modal logic. In *Proc. of Advances in Modal Logic 2000 (AiML 2000)*, 2000.

[122] Carsten Lutz and Ulrike Sattler. Mary likes all cats. In *Proc. of the 2000 Description Logic Workshop (DL 2000)*, pages 213–226. CEUR Electronic Workshop Proceedings, http://ceur-ws.org/Vol-33/, 2000.

[123] Robert MacGregor. The evolving technology of classification-based knowledge representation systems. In John F. Sowa, editor, *Principles of Semantic Networks*, pages 385–400. Morgan Kaufmann, Los Altos, 1991.

[124] Eric Mays, Robert Dionne, and Robert Weida. K-REP system overview. *SIGART Bull.*, 2(3), 1991.

[125] Marvin Minsky. A framework for representing knowledge. In J. Haugeland, editor, *Mind Design*. The MIT Press, 1981. A longer version appeared in *The Psychology of Computer Vision* (1975). Republished in [43].

[126] Ralf Molitor. Konsistenz von Wissensbasen in Beschreibungslogiken mit Rollenoperatoren. Diploma thesis, RWTH Aachen, Germany, 1997.

[127] Bernhard Nebel. Computational complexity of terminological reasoning in BACK. *Artificial Intelligence*, 34(3):371–383, 1988.

[128] Bernhard Nebel. Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43:235–249, 1990.

[129] Bernhard Nebel. Terminological cycles: Semantics and computational properties. In John F. Sowa, editor, *Principles of Semantic Networks*, pages 331–361. Morgan Kaufmann, Los Altos, 1991.

[130] Jeff Z. Pan and Ian Horrocks. Reasoning in the $\mathcal{SHOQ}(\mathcal{D}_n)$ description logic. In Ian Horrocks and Sergio Tessaris, editors, *Proceedings of the International Workshop in Description Logics 2002 (DL2002)*, volume 53 of *CEUR-WS (http://ceur-ws.org/)*, pages 53–62, 2002.

[131] Peter F. Patel-Schneider and Ian Horrocks. DLP and FaCT. In Neil V. Murray, editor, *Proc. of the Int. Conf. on Analytic Tableaux and Related Methods (TABLEAUX'99)*, volume 1617 of *Lecture Notes in Artificial Intelligence*, pages 19–23. Springer-Verlag, 1999.

[132] Christof Peltason. The BACK system — an overview. *SIGART Bull.*, 2(3):114–119, 1991.

[133] M. Ross Quillian. Word concepts: A theory and simulation of some basic capabilities. *Behavioral Science*, 12:410–430, 1967. Republished in [43].

[134] Alan Rector and Ian Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Proceedings of the Workshop on Ontological Engineering, AAAI Spring Symposium (AAAI'97)*, Stanford, CA, 1997. AAAI Press.

[135] Ulrike Sattler. A concept language extended with different kinds of transitive roles. In Günter Görz and Steffen Hölldobler, editors, *Proc. of the 20th German Annual Conf. on Artificial Intelligence (KI'96)*, volume 1137 of *Lecture Notes in Artificial Intelligence*, pages 333–345. Springer-Verlag, 1996.

[136] Ulrike Sattler. Description logics for the representation of aggregated objects. In W.Horn, editor, *Proc. of the 14th Eur. Conf. on Artificial Intelligence (ECAI 2000)*. IOS Press, Amsterdam, 2000.

[137] Andrea Schaerf. On the complexity of the instance checking problem in concept languages with existential quantification. *J. of Intelligent Information Systems*, 2:265–278, 1993.

[138] Klaus Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)*, pages 466–471, 1991.

[139] Klaus Schild. Combining terminological logics with tense logic. In *Proc. of the 6th Portuguese Conf. on Artificial Intelligence (EPIA'93)*, volume 727 of *Lecture Notes in Computer Science*, pages 105–120. Springer-Verlag, 1993.

[140] Klaus Schild. Terminological cycles and the propositional μ-calculus. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proc. of the 4th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'94)*, pages 509–520, Bonn (Germany), 1994. Morgan Kaufmann, Los Altos.

[141] Manfred Schmidt-Schauß. Subsumption in KL-ONE is undecidable. In Ron J. Brachman, Hector J. Levesque, and Ray Reiter, editors, *Proc. of the 1st Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'89)*, pages 421–431. Morgan Kaufmann, Los Altos, 1989.

[142] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with unions and complements. Technical Report SR-88-21, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern (Germany), 1988.

[143] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[144] Len K. Schubert, Randy G. Goebel, and Nicola J. Cercone. The structure and organization of a semantic net for comprehension and inference. In N. V. Findler, editor, *Associative Networks: Representation and Use of Knowledge by Computers*, pages 121–175. Academic Press, 1979.

[145] Fabrizio Sebastiani and Umberto Straccia. A computationally tractable terminological logic. In *Proc. of the 3rd Scandinavian Conf. on Artificial Intelligence SCAI-91*, pages 307–315, 1991.

[146] Valentin Shehtman. Undecidable propositional calculi. *USSR Academy of Sciences*, 75:74–116, 1982. (Russian).

[147] John F. Sowa. *Conceptuals Structures: Information Processing in Mind and Machine.* Addison Wesley Publ. Co., Reading, Massachussetts, 1984.

[148] Kent A. Spackman, Keith E. Campbell, and Roger A. Cote. SNOMED RT: A reference terminology for health care. *J. of the American Medical Informatics Association*, pages 640–644, 1997. Fall Symposium Supplement.

[149] Alfred Tarski. *A Decision Method for Elementary Algebra and Geometry.* University of California Press, Berkeley, 1951.

[150] Stephan Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation.* PhD thesis, RWTH Aachen, 2001.

[151] David Toman and Grant Weddell. On reasoning about structural equality in XML: A description logic approach. In D. Calvanese, M. Lenzerini, and R. Motwani, editors, *Proceedings of the 9th International Conference on Database Theory*, number 2572 in LNCS, pages 96–110, 2002.

[152] Johan van Benthem. Modal reduction principles. *J. of Symbolic Logic*, 41:301–312, 1976.

[153] Wiebe Van der Hoek and Maarten de Rijke. Counting objects. *J. of Logic and Computation*, 5(3):325–345, 1995.

[154] Moshe Y. Vardi. The taming of converse: Reasoning about two-way computations. In R. Parikh, editor, *Proc. of the 4th Workshop on Logics of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 413–424. Springer-Verlag, 1985.

[155] Moshe Y. Vardi and Pierre Wolper. Automata-theoretic techniques for modal logics of programs. In *Proc. of the 16th ACM SIGACT Symp. on Theory of Computing (STOC'84)*, pages 446–455, 1984.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

14

HYBRID LOGICS

Carlos Areces and Balder ten Cate

1	What are Hybrid Logics?	822
	1.1 Basic Definitions	824
2	History	826
	2.1 The Foundational Work of Prior	826
	2.2 The Sofia School	828
	2.3 Very Expressive Hybrid Languages	831
3	Model Theory	832
	3.1 Completeness	832
	3.2 Expressive Power and Characterization	837
	3.3 Interpolation and Beth Definability	844
4	Decidability and Complexity	846
5	Proof Theory	852
	5.1 Sequent Calculus	852
	5.2 Natural Deduction Calculus	854
	5.3 Tableau Calculus	855
	5.4 Resolution Calculus	857
6	Relation with Other Fields	858
7	Discussion	863

This chapter provides a modern overview of the field of hybrid logic. Hybrid logics are extensions of standard modal logics, involving symbols that name individual states in models. The first results that are nowadays considered as part of the field date back to the early work of Arthur Prior in 1951. Since then, hybrid logic has gone through a number of revivals and reinventions. Nowaways, it is a field of research in its own right, with a wealth of results, techniques, and applications.

Our main aim, in this chapter, is to provide a coherent picture of the current state of affairs in the field of hybrid logic. Rather than a comprehensive summary, we will try to give the reader a taste for the type of results and techniques that we consider hallmarks of the field. In some cases, we will only state results, with pointers to relevant literature, while in other cases we will provide full proofs.

In Section 1, we give an intuitive introduction to hybrid logics, with examples of the extra expressive power offered by the hybrid operators. This section also contains the basic definitions of syntax and semantics that are used throughout the chapter. In Section 2 we provide a short history of the field, discussing the work of Prior in the 50s, of the Sofia School in the 80s, and the work on very expressive hybrid languages in the 90s. Sections 3 and 4 form the core of

the chapter. They contain the most important techniques and results in the field, with respect to completeness, expressive power, frame definability, interpolation and complexity. In Section 5 we briefly present proof systems for hybrid languages (sequents, natural deduction, tableaux, and resolution), and we discuss some issues concerning the development of automated provers based on them. In Section 6 we comment on connections with related areas (some of which are discussed in detail in other chapters of this handbook). Section 7 finishes the chapter with a summary and general perspectives.

1 WHAT ARE HYBRID LOGICS?

In their simplest form, hybrid languages are modal languages that have special symbols to name individual states in models. These new symbols, which are called *nominals*, enter the stage gracefully: we simply add a new sort of atomic symbols $\mathsf{NOM} = \{i, j, k, \ldots\}$ disjoint from the set PROP of propositional variables and let them combine freely in formulas. For example, if i is a nominal and p and q are propositional variables, then

$$\Diamond(i \wedge p) \wedge \Diamond(i \wedge q) \to \Diamond(p \wedge q), \tag{1}$$

is a well formed formula. Now for the important twist: since nominals name individual states in the model, they denote *singleton sets*. In other words, they are true at a unique point in the model. Once this step has been taken, the whole landscape changes. For example, (1) becomes a validity: let $\mathcal{M}$ be a model, m a state in the domain of $\mathcal{M}$, and suppose $\mathcal{M}, m \models \Diamond(i \wedge p) \wedge \Diamond(i \wedge q)$. Then some successor state m' of m satisfies $i \wedge p$, and some successor state m'' of m satisfies $i \wedge q$. Since i is a nominal, it is true at a unique point in $\mathcal{M}$. Hence $m' = m''$ and we have $\mathcal{M}, m \models \Diamond(p \wedge q)$. Note that (1) could be falsified if i were an ordinary propositional variable.

When we realize the potential that nominals have, an interesting idea suggests itself: to introduce, for each nominal i, an operator $@_i$ that allows us to jump to the point named by i. The formula $@_i\varphi$ (read "at i, φ") moves the point of evaluation to the state named by i and evaluates φ there. These operators satisfy many nice logical properties. For a start, each $@_i$ is a normal modal operator: it satisfies the distributivity axiom $(@_i(\varphi \to \psi) \to (@_i\varphi \to @_i\psi))$ and the necessitation rule (if φ is valid, then $@_i\varphi$ is also valid). Moreover, it is self-dual: $@_i\varphi$ is equivalent to $\neg @_i \neg\varphi$. In an intuitive sense, the $@_i$ operators provide a bridge between semantics and syntax by internalizing the satisfaction relation '$\models$' into the logical language:

$$\mathcal{M}, w \models \varphi \;\text{ iff }\; \mathcal{M} \models @_i\varphi, \;\text{where } i \text{ is a nominal naming } w.$$

For this reason, these operators are usually called *satisfaction operators*.

Aiming to make full use of the flexibility provided by direct reference to specific points in the model naturally leads to further enrichment of the language. One possibility would be to have not only names for individual states but also variables ranging over states, with corresponding quantifiers. We would then be able to write formulas like

$$\forall y. \Diamond y. \tag{2}$$

The first-order translation of this formula is $\forall y. \exists z.(R(x, z) \wedge z = y)$ or, simply, $\forall y. R(x, y)$, forcing the current state to be related to all states in the domain. The $\forall$ quantifier is very expressive. As discussed in [32], even the basic modal language extended with state variables and this universal quantifier is undecidable. Moreover, $\forall$ and $@$ together give us already full first-order

expressive power (cf. Section 3.2). Nevertheless, the $\forall$ quantifier is historically important. The earliest treatments are probably those of [117, 118, 46].

The $\forall$ quantifier is very "classical." If we think modally, and remember that evaluation of modal formulas takes place *at a given point*, a different kind of binder suggests itself. The $\downarrow$ binder binds variables to points but, unlike $\forall$, it binds to the *current* point. In essence, it enables us to create a name for the here-and-now, and refer to it later in the formula. For example, the formula

$$\downarrow y.\Diamond y \tag{3}$$

is true at a state m iff m is related to itself. The intuitive reading of (3) is quite straightforward: the formula says "call the current state y and check that y is reachable." The difference between $\forall$ and $\downarrow$ is subtle, but important. $\forall$ is *global*, in the sense that formulas containing $\forall$ are not preserved under generated submodels [32]. On the other hand, $\downarrow$ is intrinsically local and, as we will show in Theorems 16 and 18, it can be characterized in terms of the operation of taking generated submodels.

Like $\forall$, the $\downarrow$ binder has been invented independently on several occasions. For example, in [122], $\downarrow$ is introduced as part of an investigation into temporal semantics and temporal databases, [131] uses it to aid reasoning about automata, it is related to the freeze operator in [88], and [52] employs it as part of his treatment of indexicality. However, none of the systems just mentioned allows the free syntactic interplay of variables with the underlying propositional logic; that is, they make use of $\downarrow$, but in languages that are not fully hybrid. The earliest paper to introduce it into a fully hybrid language seems to be [77].

Note that satisfaction operators work in perfect coordination with $\downarrow$. Whereas $\downarrow$ "stores" the current point of evaluation (by binding a variable to it), the satisfaction operators enable us to "retrieve" stored information by shifting the point of evaluation in the model. By using the "storing and retrieving" intuition it is easy to define complex properties. For example, Kamp's temporal *until* operator U (with semantics: $U(\varphi, \psi)$ is true at a state m if there is a future state m' where φ holds, such that ψ holds in all states between m and m') can be defined as follows:

$$U(\varphi, \psi) := \downarrow x.\Diamond\downarrow y.(\varphi \wedge @_x\Box(\Diamond y \rightarrow \psi)).$$

Let us see how this work. First, we name the current state x using $\downarrow$, and use the $\Diamond$ operator to find a suitable successor state, which we call y, where φ holds. Without the @ operator we would be stuck in that successor state, but we can use @ to go back to x and demand that in all successors of x having y as a successor, ψ holds.

Summarizing the above discussion, we can say that the term *hybrid logic* refers to a family of extensions of the basic hybrid language with devices that, in one way or another, allow for explicit reference to individual states of the Kripke model. But, why are hybrid logics called *hybrid*?

One explanation comes from the work of Arthur Prior in the 1950s. As we will discuss more in detail in Section 2, Prior was interested in the relation between what McTaggart called the A-series and B-series of time [109]. Following McTaggart's analysis of time in terms of the A-series of past, present and future and the B-series of earlier and later, Prior discusses two logical systems: the I-calculus aims to capture the properties of the B-series and takes variables ranging over instants as primitive, while the T-calculus examines tenses and takes variables ranging over propositions. In [117, Chapter V.6], Prior proposes a way to develop the I-calculus *inside* the T-calculus, and for this he allows instant-variables to be used together with propositional variables. He will call this step "the third grade of tense-logical involvement" in [118, Chapter XI], where instant-variables are treated as representing (special) propositions. From this perspective, the

terms hybrid applies to the "confusion" of terms (the variables over instants) with formulas (the propositional variables).

There is another sense in which hybrid logics are hybrid, namely that, both in terms of expressive power and in terms of the techniques used to analyze them, hybrid languages lie in between the basic modal language and first-order logic. While having a distinctly modal flavor, hybrid logics enjoy features which are of a clear first-order nature. As we discussed above, the more expressive hybrid languages include binders and variables over elements of the domain, traditional hallmarks of first-order languages, while nominals are nothing else than first-order constants. The nominals and satisfaction operators also introduce a restricted form of equality: a state m in a model can satisfy a nominal i if and only if it is equal to the denotation of i, and a model $\mathcal{M}$ satisfies $@_i j$ if and only if the denotations of i and j coincide. In other words, nominals introduce equality between the point of evaluation and a named state, while satisfaction operators enable us to express equality between named states. Concerning first-order techniques which can be used for hybrid languages, we will see in Section 3.1 for example, that nominals can be used as 'witnesses' in a classical Henkin-style completeness proof for hybrid languages, and classical first-order notions like potential isomorphisms are useful for characterizing the expressive power of hybrid languages. In Section 3.3, we will see a very general interpolation result, the proof of which relies on the fact that shared nominals can be "bound away" using $\downarrow$, in the same way that shared constants can be replaced by existentially quantified variables in first-order logic.

For a more detailed introduction, including further intuitive examples using the different hybrid languages, the reader is referred to [26]. The Hybrid Logic Web Pages [3] provides additional information and a broad on-line bibliography. We now move on to the basic definitions of syntax and semantics that will be used through the chapter.

1.1 Basic Definitions

The simplest hybrid language is $\mathcal{H}$, which extends the basic modal language with nominals only. Further extensions will be named by listing the additional operators. The most expressive system we will discuss in detail is $\mathcal{H}(\mathsf{E}, @, \downarrow)$, with the existential modality E, @-operators, and the $\downarrow$ binder (when considering languages containing the $\downarrow$ binder, it is implicitly understood that the language also contains state variables). At various points, we will briefly mention other hybrid languages as well (e.g., hybrid extensions of temporal and dynamic logics).

The following two definitions give the syntax and semantics of $\mathcal{H}(\mathsf{E}, @, \downarrow)$. The corresponding definitions for sublanguages of $\mathcal{H}(\mathsf{E}, @, \downarrow)$ can be obtained by leaving out irrelevant clauses.

DEFINITION 1. Let $\mathsf{REL} = \{R_1, R_2, \ldots\}$ (the *relational symbols*), $\mathsf{PROP} = \{p_1, p_2, \ldots\}$ (the *propositional variables*), $\mathsf{NOM} = \{i_1, i_2, \ldots\}$ (the *nominals*), and $\mathsf{SVAR} = \{x_1, x_2, \ldots\}$ (the *state variables*) be pairwise disjoint, countably infinite sets of symbols. By a *state symbol*, we will mean any element of $\mathsf{NOM} \cup \mathsf{SVAR}$. The well-formed formulas of the hybrid language $\mathcal{H}(\mathsf{E}, @, \downarrow)$ in the signature $\langle \mathsf{REL}, \mathsf{PROP}, \mathsf{NOM}, \mathsf{SVAR} \rangle$ are given by the following recursive definition:

$$\mathsf{FORMS} \;::=\; \top \mid p \mid s \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle R \rangle \varphi \mid \mathsf{E}\varphi \mid @_s\varphi \mid \downarrow x.\varphi,$$

where $p \in \mathsf{PROP}$, $s \in \mathsf{NOM} \cup \mathsf{SVAR}$, $x \in \mathsf{SVAR}$, $R \in \mathsf{REL}$ and $\varphi, \varphi_1, \varphi_2 \in \mathsf{FORMS}$.

Given a set of formulas $\Gamma \subseteq \mathsf{FORMS}$, we will use $\mathsf{PROP}(\Gamma)$, $\mathsf{NOM}(\Gamma)$ and $\mathsf{SVAR}(\Gamma)$ to denote, respectively, the set of propositional variables, nominals, and state variables occurring in formulas in Γ. Also, for φ a formula, $\mathsf{SF}(\varphi)$ will be the set of subformulas of φ.

Note that the above syntax is simply that of ordinary (multi-modal) propositional modal logic extended with clauses for the state symbols and for $\mathsf{E}\varphi$, $@_s\varphi$ and $\downarrow x_j.\varphi$. Also, note that, like propositional variables, nominals and state variables can be used as atomic formulas. The difference between nominals and state variables is analogous to the difference between constants and variables in first-order logic: nominals cannot be bound by $\downarrow$, and their interpretation is specified by the model, whereas state variables are interpreted by assignment functions, and they can be bound by the $\downarrow$-binder.

The notions of *free* and *bound* state variable are defined as in first-order logic, with $\downarrow$ as the only binding operator. Similarly, other syntactic notions (such as *substitution*, and a state symbol t being *substitutable for* x *in* φ) are defined as in first-order logic. A *sentence* is a formula containing no free state variables. Furthermore, a formula is *pure* if it contains no propositional variables, and *nominal-free* if it contains no nominals.

In the remainder of the chapter we will assume fixed a signature $\langle \mathsf{REL}, \mathsf{PROP}, \mathsf{NOM}, \mathsf{SVAR} \rangle$. Now for the semantics.

DEFINITION 2. A (hybrid) *model* $\mathcal{M}$ is a triple $\mathcal{M} = \langle M, (R^{\mathcal{M}})_{R\in\mathsf{REL}}, V \rangle$ such that M is a non-empty set, each $R^{\mathcal{M}}$ is a binary relation on M, and $V : \mathsf{PROP} \cup \mathsf{NOM} \to \wp(M)$ is such that for all nominals $i \in \mathsf{NOM}$, $V(i)$ is a singleton subset of M. We usually write M (roman letters) for the domain of a model $\mathcal{M}$, and call the elements of M *states*, *worlds* or *points*. Each $R^{\mathcal{M}}$ is an *accessibility relation*, and V is the *valuation*. A *frame* is defined in the usual way: as a model without a valuation. If $\mathcal{F} = \langle M, (R^{\mathcal{F}})_{R\in\mathsf{REL}} \rangle$ is a frame and V is a valuation on M, then $\mathcal{M} = \langle \mathcal{F}, V \rangle$ is the model $\langle M, (R^{\mathcal{F}})_{R\in\mathsf{REL}}, V \rangle$. In this case we, say that $\mathcal{M}$ *is based on* $\mathcal{F}$, and that $\mathcal{F}$ *is the underlying frame of* $\mathcal{M}$.

An *assignment* g for $\mathcal{M}$ is a mapping $g : \mathsf{SVAR} \to M$. Given an assignment $g : \mathsf{SVAR} \to M$, a state variable $x \in \mathsf{SVAR}$, and a state $m \in M$, we define g^x_m (an x*-variant of* g) by letting $g^x_m(x) = m$ and $g^x_m(y) = g(y)$ for all $y \neq x$.

Let $\mathcal{M} = \langle M, (R^{\mathcal{M}})_{R\in\mathsf{REL}}, V \rangle$ be a model, $m \in M$, and g an assignment for $\mathcal{M}$. For any state symbol $s \in \mathsf{NOM} \cup \mathsf{SVAR}$, let $[s]^{\mathcal{M},g}$ be the state denoted by s (i.e., for $i \in \mathsf{NOM}$, $[i]^{\mathcal{M},g}$ is the unique $m \in M$ such that $V(i) = \{m\}$, and for $x \in \mathsf{SVAR}$, $[x]^{\mathcal{M},g} = g(x)$). Then the *satisfaction relation* is defined as follows:

$$
\begin{array}{lll}
\mathcal{M}, g, m \models \top & & \\
\mathcal{M}, g, m \models p & \text{iff} & m \in V(p) \quad \text{for } p \in \mathsf{PROP} \\
\mathcal{M}, g, m \models s & \text{iff} & m = [s]^{\mathcal{M},g} \quad \text{for } s \in \mathsf{NOM} \cup \mathsf{SVAR} \\
\mathcal{M}, g, m \models \neg\varphi & \text{iff} & \mathcal{M}, g, m \not\models \varphi \\
\mathcal{M}, g, m \models \varphi_1 \wedge \varphi_2 & \text{iff} & \mathcal{M}, g, m \models \varphi_1 \text{ and } \mathcal{M}, g, m \models \varphi_2 \\
\mathcal{M}, g, m \models \langle R \rangle\varphi & \text{iff} & \text{there is a state } m' \text{ such that } R^{\mathcal{M}}(m, m') \text{ and } \mathcal{M}, g, m' \models \varphi \\
\mathcal{M}, g, m \models \mathsf{E}\varphi & \text{iff} & \text{there is a state } m' \in M \text{ such that } \mathcal{M}, g, m' \models \varphi \\
\mathcal{M}, g, m \models @_s\varphi & \text{iff} & \mathcal{M}, g, [s]^{\mathcal{M},g} \models \varphi \quad \text{for } s \in \mathsf{NOM} \cup \mathsf{SVAR} \\
\mathcal{M}, g, m \models \downarrow x.\varphi & \text{iff} & \mathcal{M}, g^x_m, m \models \varphi.
\end{array}
$$

The first six clauses in the definition of the satisfaction relation are similar to the ones for the basic modal language, except that they are relativized to an additional assignment function. Recall that nominals and state variables can be used as atomic formulas, in which case they act as propositional variables that are true at a unique state. The $\downarrow$ binder binds state variables to the state where evaluation is being performed (the *current world*), and $@_s$ shifts evaluation to the state named by s. As in first-order logic, if φ is a *sentence* (i.e., a formula with no free state variables), the truth of φ at a state in a model does not depend on the assignment. Hence, in this case we will write $\mathcal{M}, m \models \varphi$ instead of $\mathcal{M}, g, m \models \varphi$.

A formula φ is said to be *globally true* in a model $\mathcal{M}$ under an assignment g (notation: $\mathcal{M}, g \models \varphi$), if $\mathcal{M}, g, m \models \varphi$ for all $m \in M$. A formula φ is *satisfiable* if there is a model $\mathcal{M}$, an assignment g on $\mathcal{M}$, and a world $m \in M$ such that $\mathcal{M}, g, m \models \varphi$. A formula φ is *valid* (notation: $\models \varphi$) if for all models $\mathcal{M}$ and assignments g, $\mathcal{M}, g \models \varphi$. A formula φ is a *local consequence* of a set of formulas Σ if for all models $\mathcal{M}$, assignments g, and points $m \in M$, $\mathcal{M}, g, m \models \Sigma$ implies $\mathcal{M}, g, m \models \varphi$. A formula φ is a *global consequence* of a set of formulas Σ if for all models $\mathcal{M}$ and assignments g, $\mathcal{M}, g \models \Sigma$ implies $\mathcal{M}, g \models \varphi$. We denote local consequence by $\Sigma \models^{loc} \varphi$ and global consequence by $\Sigma \models^{glo} \varphi$. As in ordinary propositional modal logic, local consequence is strictly stronger than global consequence.

Definitions 1 and 2 specify the syntax and semantics of the most expressive hybrid language we are going to discuss in detail, $\mathcal{H}(\mathsf{E}, @, \downarrow)$. Two important fragments of this language are $\mathcal{H}(@, \downarrow)$, which is obtained by dropping the clauses for the existential modality E, and $\mathcal{H}(@)$, which is obtained by dropping in addition the state variables and the $\downarrow$-binder. In other words, $\mathcal{H}(@)$ is simply the extension of the basic modal language with nominals and satisfaction operators. The languages $\mathcal{H}(@)$ and $\mathcal{H}(@, \downarrow)$ will receive most attention in this chapter.

2 HISTORY

In this section we will provide an overview of the historical development of hybrid languages, starting with the pioneering work of Prior, through the "revival" in the late eighties and early nineties in Sofia, and ending with the work of Blackburn and Seligman in the late nineties.

2.1 The Foundational Work of Prior

The work of Prior in modal logic and in particular in the modal analysis of time is well known, to the point that he is usually regarded as the inventor of temporal logic. For a detailed discussion of Prior's contributions to this field, together with some biographical information, see [111]. The following discussion is based on [51], a short but very good overview. See also [27], especially Section 4.

Prior is considered one of the most important promoters of the application of modal syntax to the formalisation of a wide variety of phenomena. Less well known is the fact that Prior, in collaboration with Carew Meredith, devised a version of possible worlds semantics roughly at the same time as, but independently of, the work of Carnap on modal semantics and several years before Kripke published his first paper on the topic. Interestingly, this part of Prior's work is already closely related to hybrid logic.

Nowadays, the view that modal logic can be seen as a fragment of first-order or second-order logic is commonplace. This is fairly straightforwards once we observe the possible worlds semantics of modal operators. When reading the earlier work of Prior, however, we should keep in mind that, at that time, most modal intuitions came solely from axiomatics. Nevertheless, in Prior's (unpublished) second book "The Craft of Formal Logic" (completed in 1951) we can find the following passage:

> For the similarity in behaviour between signs of modality and signs of quantity, various explanations may be offered. It may be, for example, that signs of modality are just ordinary quantifiers operating upon a peculiar subject-matter, namely possible states of affairs. . . It would not be quite accurate to describe theories of this

> sort as "reducing modality to quantity." They do reduce modal *distinctions* to distinctions of quantity, but the variables to which the quantifiers are attached retain something modal in their signification — they signify "possibilities", "chances", "possible states of affairs", "possible combinations of truth-values", or the like.

Two things should be noticed in this passage. Firstly, the reference to "possible states of affairs" and even "possible combinations of truth-values," is a very early reference to possible worlds semantics. Secondly, note Prior's strong reservations concerning "reducing modality to quantity". This early intuition on the foundational nature of modality later grew into a mature philosophy in Prior's view that quantification over possible worlds and instants was to be interpreted in terms of modality and tense — which constituted primitive notions — and not vice versa (although he did recognized that the study of both quantity and modality could benefit of each other).

Three years later, in 1954 at the New Zealand Congress of Philosophy, Prior presented a paper (not published until much later as [116]) in which his philosophical position is made more explicit. Working already in the framework of temporal logic, he introduces in this paper the I-calculus (which he will later call the U-calculus). In the I-calculus, propositions of the tense calculus are treated as predicates expressing properties of dates (which are represented by variables). The formula px should be read as "p at x," and I is a binary relation taking dates as arguments where Ixy is read as "y is later than x." Using an arbitrary date x to represent the time of utterance, Fp (intuitively, "the proposition p happens in the future") is equated with $\exists y.(Ixy \wedge py)$ (i.e., "p at some time later than x") and similarly for Pp, "the proposition p happens in the past." Prior mentions already that, by imposing various conditions on the relation I, analogues of the axioms of the tense calculus can be derived in the I-calculus.

Later in the same paper, Prior includes a detailed warning against regarding this interpretation of the tense calculus within the I-calculus as "a metaphysical explanation of what we mean by *is*, *has been* and *will be*"; he stresses that the I-calculus is not "metaphysically fundamental." He explains that F(Socrates is sitting down) means "It is *now* the case that it will be the case that Socrates is sitting down," and there is no genuine way of representing the indexical *now* in the I-calculus (he says that the free variable x is "a complete sham"). He continues: "If there is to be any 'interpretation' of our calculi in the metaphysical sense, it will probably need to be the other way round; that is, the I-calculus should be exhibited as a logical construction out of the PF-calculus rather than *vice versa*." This idea of the primacy of the tense calculus over the I-calculus — or, as he was later to put it, of McTaggart's A-series over the B-series, see [109] — was to become a central and distinctive tenet of his philosophy. These issues form the theme of his final, unfinished, book [119], but they already appear in some earlier articles.

But of course, the reconstruction of the I-calculus within the tense calculus is impossible, as the I-calculus is strictly more expressive than the tense calculus. Prior recognized this fact and investigated ways to extend the expressive power of the tense calculus to permit the reconstruction. This directly led to what we call today *very expressive hybrid languages* (i.e., hybrid languages including the $\forall$ binder). In [117, Chapter V.6], he actually proposes a way to develop the I-calculus inside the tense calculus, and for this he allows instant variables to be used together with propositional variables. He will call this step "the third grade of tense-logical involvement" in [118, Chapter XI], where instant variables are treated as representing (special) propositions.

We see, then, that Prior's development of hybrid languages was rooted in his philosophical convictions, and was instrumental in the implementation of some of his very early intuitions on time and tense. Prior's death in 1969 put an end to these investigations. Notice though, that Prior was never fully satisfied with his solution. It was technically correct (and actually quite bold and ingenious) but he was concerned that, in managing to "upgrade" the tense calculus to full

first-order expressivity, the language had lost its claim to a metaphysical fundamentality. Robert Bull, a student of Prior, pushed the ideas of hybridization further in [46], where he provides an axiomatization and completeness result for a logic containing variables for *paths* on a model, which he calls "history-propositional" variables.

2.2 The Sofia School

As we saw, the roots of hybrid logic go back to Prior and Bull. About fifteen years later in Sofia, Bulgaria, nominals were re-discovered by Gargov, Passy and Tinchev in their investigations on Boolean modal logic and propositional dynamic logic. One of the issues that led them into these investigations was the following asymmetry in the expressive power of the modal language. The union of two accessibility relations is definable in the basic modal language, in the sense that the formula

$$\langle T\rangle p \leftrightarrow \langle R\rangle p \vee \langle S\rangle p$$

is valid on a frame precisely if the accessibility relation interpreting $\langle T\rangle$ is the union of the accessibility relations interpreting $\langle R\rangle$ and $\langle S\rangle$. Moreover, when added to the basic modal language, this formula completely axiomatizes the modal logic of the relevant class of frames.

Surprisingly, *intersection* of accessibility relations is not definable in the same way: it follows from the Goldblatt-Thomason theorem [76] that there is no formula in the basic modal language that is valid on a frame precisely if the accessibility relation of $\langle T\rangle$ is the intersection of the accessibility relation of $\langle R\rangle$ and $\langle S\rangle$. And even though the axiom scheme $\langle T\rangle p \rightarrow (\langle R\rangle p \wedge \langle S\rangle p)$ (together with the standard axioms and rules for the basic polymodal logic) completely axiomatizes the logic of this frame class, it is valid on the larger class where the accessibility relation of $\langle T\rangle$ is contained in the intersection of the accessibility relation of $\langle R\rangle$ and $\langle S\rangle$.

Now, Gargov, Passy and Tinchev showed in [74] that intersection *can* be defined using *nominals*. Indeed, for i a nominal, the axiom scheme

$$\langle T\rangle i \leftrightarrow \langle R\rangle i \wedge \langle S\rangle i$$

defines intersection in the above sense, and exactly axiomatizes the logic of the relevant class of frames (when added to an appropriate base axiomatization)[1]. The same story goes for complementation: there is no formula of the basic modal language that is valid on a frame precisely if the accessibility relation of $\langle R\rangle$ is the complement of the accessibility relation of S, but such a formula exists when nominals are added to the language: $\langle R\rangle i \leftrightarrow \neg\langle S\rangle i$.

This form of capturing the Boolean operations (together with an alternative based on the "sufficiency operator" $\boxminus$) was investigated by Gargov, Passy and Tinchev in [74]. In that paper, the first complete axiomatization of the minimal hybrid language is given. Following [75], recursively define $\Box$- and $\Diamond$-forms as follows: 1) \$ is both a $\Box$- and a $\Diamond$-form (where \$ is a fixed symbol not occurring in the language); 2) If L is a $\Box$-form and φ a formula, then $(\varphi \rightarrow L)$ and $\Box L$ are also $\Box$-forms; and 3) If M is a $\Diamond$-form and φ is a formula, then $(\varphi \wedge M)$ and $(\Diamond M)$ are also $\Diamond$-forms. For F a $\Box$- or $\Diamond$-form and φ a formula, let $F(\varphi)$ be the formula obtained by replacing the unique occurrence of \$ in F by φ. Now, Gargov, Passy and Tinchev showed that any complete axiomatization of the basic modal language, extended with the axioms

$$M(i \wedge \varphi) \rightarrow L(i \rightarrow \varphi) \quad \text{for } i \text{ a nominal, } L \text{ a } \Box\text{-form and } M \text{ a } \Diamond\text{-form}$$

completely axiomatizes the hybrid logic (in the language $\mathcal{H}$) of the class of all frames.

[1] Note that this implies that the Goldblatt-Thomason theorem, in its usual form, does not hold for hybrid languages.

Axiom Schemes:

(A0)	All propositional tautologies
(A1)	$\langle\nu\rangle i$
(A2)	$\langle\nu\rangle(i \wedge \varphi) \to [\nu](i \to \varphi)$
(A3)	$\varphi \to \langle\nu\rangle\varphi$
(A4)	$\langle\nu\rangle\langle\nu\rangle\varphi \to \langle\nu\rangle\varphi$
(A5)	$\varphi \to [\nu]\langle\nu\rangle\varphi$
(A6)	$\langle\alpha\rangle\varphi \to \langle\nu\rangle\varphi$
(A7)	$\langle\alpha\beta\rangle\varphi \to \langle\alpha\rangle\langle\beta\rangle\varphi$
(A8)	$\langle\alpha \cup \beta\rangle i \leftrightarrow \langle\alpha\rangle i \vee \langle\beta\rangle i$
(A9)	$\langle\alpha \cap \beta\rangle i \leftrightarrow \langle\alpha\rangle i \wedge \langle\beta\rangle i$
(A10)	$\langle\bar{\alpha}\rangle i \leftrightarrow [\alpha]\neg i$
(A11)	$\alpha \subset \beta \leftrightarrow [\alpha \cap \bar{\beta}]\bot$
(A12)	$\langle\nu\rangle(i \wedge \langle\alpha^{-1}\rangle j) \leftrightarrow \langle\nu\rangle(j \wedge \langle\alpha\rangle i)$
(A13)	$\langle\varphi?\rangle\psi \leftrightarrow \varphi \wedge \psi$
(A14)	$\langle\alpha^*\rangle\varphi \leftrightarrow \varphi \vee \langle\alpha\rangle\langle\alpha^*\rangle\varphi$
(A15)	$[\alpha](\varphi \to \psi) \to ([\alpha]\varphi \to [\alpha]\psi)$

Rules:

(R1)	If $\vdash [\alpha]\neg i$ for some i not in α, then $\vdash [\alpha]\bot$.
(R2)	If $\vdash [\beta][\alpha^n]\varphi$ for all $n \in \mathbb{N}$, then $\vdash [\beta][\alpha^*]\varphi$.
(R3)	If $\vdash \varphi$, then $\vdash [\nu]\varphi$.
(R4)	If $\vdash \varphi$ and $\vdash \varphi \to \psi$, then $\vdash \psi$

Where φ, ψ are formulas, α, β programs, ν the universal program and i, j nominals.

Figure 1. Axiomatization of CPDL$(\cap, ^-, \subset, ^{-1})$

Besides the minimal hybrid language $\mathcal{H}$, Gargov, Passy and Tinchev also studied a richer hybrid language, obtained by extending propositional dynamic logic (PDL, cf. Chapter 12 of this handbook) with nominals. Intersection of accessibility relations is particularly interesting in this setting, as it can be interpreted as parallelism, or concurrency of programs. Passy and Tinchev [113] propose an extension of PDL with nominals and the universal modality, which they call Combinatory PDL (CPDL). The paper contains an axiomatization of CPDL$(\cap, ^-, \subset, ^{-1})$, combinatory PDL extended with program intersection, complementation, subprograms and inverse, shown in Figure 1. Note that this axiomatization contains an infinitary rule (R2), i.e., an inference rule with infinitely many premises.

Besides the standard axioms and rules of PDL, and the axioms for the universal program ν, notice the definitions of union (A8), intersection (A9), complement (A10), subprogram (A11) and inverse program (A12). Notice also how the presence of the universal program ν helps defining the behaviour of nominals in axioms (A1) and (A2). Finally, notice the "Gabbay-Burgess-style rule" (R1) [67], which ensures that models are named, i.e., each state in the model is the denotation of some nominal (this also implies that models are countable). Axiomatizations for sublanguages of CPDL$(\cap, ^-, \subset, ^{-1})$ are obtained by dropping the corresponding definitions of the absent operators. In particular CPDL, "core" combinatory PDL, is axiomatized by axioms

(A1) to (A8), (A13) to (A15) and rules (R1) to (R4)[2].

Passy and Tinchev proved a number of interesting properties of CPDL (see [115] for further details). For example, they observed that named models (i.e., models in which each state is named by a nominal) can be completely described by a set of formulas of the form $(\neg)@_i p$, $(\neg)@_i\Diamond j$ or $(\neg)@_i j$. Clearly, this property only depends on the expressive power of nominals and @, and hence holds already for $\mathcal{H}(@)$. This observation provides the theoretical basis for automated theorem proving and model building via the definition of Herbrand models (i.e., a model can be represented by the set of elementary formulas which are true in it, see [17]).

With respect to (un)decidability results, naturally the negative results concerning the undecidability of both global and local consequence in PDL [85] transfers to CPDL. Passy and Tinchev provide some (un)decidability results for satisfiability of languages related to CPDL in [115], while Gargov provides in [71] a finitary axiomatization of CPDL and proves the finite model property and decidability of the satisfiability problem for CPDL. Actually, the complexity of satisfiability in CPDL coincides with the one in PDL, EXPTIME-complete [56, 55].

THEOREM 3. *For $\Gamma \cup \{\varphi\}$ a decidable set of CPDL formulas, deciding whether $\Gamma \models^{glo} \varphi$ and $\Gamma \models^{loc} \varphi$ is Π^1_1-complete. On the other hand, satisfiability of CPDL formulas is* EXPTIME-*complete.*

Gargov's axiomatizability result mentioned above uses Segerberg's axiom $\varphi\wedge[\alpha^*](\varphi \to [\alpha]\varphi) \to [\alpha^*]\varphi$ to replace the (R2) rule and shows that the (R1) rule is redundant, but infinitary rules cannot always be eliminated. For example, satisfiability of CPDL($^-$) is highly undecidable (Σ^1_1-complete) from which it follows that no finitary axiomatization can be complete. Passy and Tinchev [115] discuss the issue of eliminability of the infinitary rules in detail (cf. also [98] for more recent results on infinitary axiomatizations of hybrid logics).

We now move into more expressive hybrid languages similar to those used by Prior and Bull. Chapter III of [115] is devoted to CDL, Combinatory Dynamic Logic which allows quantification over state variables. Interestingly, the authors seem to present CDL as an alternative to quantified modal logic, stating that replacing classical quantification (over the domains in each state of the model) by hybrid quantification (over the states themselves) leads to a better behaved system. While this is true, it also leads to a system which does not resemble quantified modal logic! In any case, it is interesting to see that, once nominals have been discovered, explicit quantification over states becomes a natural extension.

The following complete axiomatization of CDL is given in [115]:

	All axioms and rules of CPDL minus (R1), plus
(A16)	$\exists c.c$
(A17)	$\forall c.\varphi \to \varphi[c/d]$
(A18)	$\forall c.[\alpha]\varphi \to [\alpha]\forall c.\varphi$ for c with no free occurrences in α.
(R5)	If $\vdash \varphi$, then $\vdash \forall c.\varphi$.

The Sofia tradition in hybrid logics continues with the work of Goranko. In [72], Gargov and Goranko investigate the basic modal language extended first with nominals and the universal and existential modalities ($\mathcal{H}(\mathsf{E})$), and then with the difference operator D ($\mathcal{ML}(\mathsf{D})$)[3]. They prove that both languages are equivalent with respect to frame definability, and then provide characterizations of frame definability for these languages.

[2] Actually, in [115], the infinitary version of (R1) "If $\vdash [\alpha]\neg i$ for all $i \in \mathsf{NOM}$ then $\vdash [\alpha]\bot$" is discussed, which is necessary for completeness in some extensions of CPDL.

[3] The semantic condition for the difference operator D is $\mathcal{M}, w \models \mathsf{D}\varphi$ iff there is a $w' \neq w$ such that $\mathcal{M}, w' \models \varphi$.

The work of Gargov and Goranko is historically relevant because, within the Sofia school, it marks the start of research on hybrid logics as such, and not as part of their research on extensions of PDL. Around the same time, but independently, Blackburn was studying simple hybrid languages over a Prior-style tense logic [21, 22]. These two lines of research can be considered the origins of the current perspective on hybrid logics.

Goranko is also the first to investigate the $\downarrow$ binder in the context of hybrid logic. In [77], he extends the basic modal language with the universal modality and the $\downarrow$ binder with only a single state variable (though using a slightly different notation). Goranko provides an axiomatization for this logic, and illustrations of its high expressivity (sufficient, for example, to define Kamp's $U(p, q)$ and $S(p, q)$ and Stavi's $U'(p, q)$ and $S'(p, q)$ temporal operators and to simulate Prior's instant variables), and shows that the satisfiability problem for this language is undecidable. He mentions in the same paper that introducing multiple state variables would be possible, and investigates the resulting language in more detail in [78].

In [79], Goranko uses hybrid binders to design CTL$_{rp}$ (CTL with reference pointers), a computation tree logic for finitely branching ω^+-trees, and defines syntactic and semantic interpretations between CTL* and CTL$_{rp}$. In particular, this yields a complete axiomatization for the translations of all valid CTL*-formulas, a step forwards in the search for a complete direct axiomatization of CTL*, a long standing open problem finally solved in [121].

With this we conclude our (necessarily brief) overview of the work on hybrid logics done by the Sofia School. It is interesting to note that most of the languages studied by the Sofia school included the universal modality. In the following years and mainly through the work of Blackburn and Seligman, research in hybrid languages deals with, on the one hand, weak languages containing only nominals (e.g., [23, 33]) and, on the other hand, very expressive languages containing binders (e.g., [31, 35, 37]).

2.3 Very Expressive Hybrid Languages

In the mid-nineties, Blackburn and Seligman [31] studied a number of very expressive hybrid languages, obtained by means of various state variable binders. We will review a few of these binders here, most of which will not return in the remainder of the chapter.

Up to now, we have introduced two hybrid binders, the "classical" $\exists$ and the "more modal" $\downarrow$. Let us review their semantic definitions. Given a model $\mathcal{M} = \langle M, (R^{\mathcal{M}})_{R \in \mathsf{REL}}, V\rangle$, an assignment g in $\mathcal{M}$ and $m \in M$:

$$\begin{array}{lll} \mathcal{M}, g, m \models \exists x.\varphi & \text{iff} & \mathcal{M}, g^x_{m'}, m \models \varphi \text{ for some } m' \in M. \\ \mathcal{M}, g, m \models \downarrow x.\varphi & \text{iff} & \mathcal{M}, g^x_m, m \models \varphi. \end{array}$$

Both quantifiers let us change the value assigned to x, without changing the point of evaluation. In [31] Blackburn and Seligman investigate two other binders which, besides changing the value of the bound variable, also change the point of evaluation:

$$\begin{array}{lll} \mathcal{M}, g, m \models \Sigma x.\varphi & \text{iff} & \mathcal{M}, g^x_{m'}, m' \models \varphi \text{ for some } m' \in M. \\ \mathcal{M}, g, m \models \Downarrow x.\varphi & \text{iff} & \mathcal{M}, g^x_m, m' \models \varphi \text{ for some } m' \in M. \end{array}$$

It is not hard to see that $\Sigma x.\varphi$ is equivalent to $\mathsf{E}{\downarrow} x.\varphi$, whereas $\Downarrow x.\varphi$ is equivalent to ${\downarrow} x.\mathsf{E}\varphi$. The Standard Translation (cf. Chapter 1 of this handbook) may be extended to these hybrid languages, in which case the appropriate clauses for these operators would be as follows (we provide also

the clause for E for comparison):

$$
\begin{array}{lcl}
ST_x(\mathsf{E}\varphi) & = & \exists z.ST_z(\varphi) \quad (z \text{ a variable not in } \varphi) \\
ST_x(\exists y.\varphi) & = & \exists y.ST_x(\varphi) \\
ST_x(\downarrow y.\varphi) & = & \exists y.(y = x \wedge ST_x(\varphi)) \\
ST_x(\Sigma y.\varphi) & = & \exists y.ST_y(\varphi) \\
ST_x(\Downarrow y.\varphi) & = & \exists z.\exists y.(y = x \wedge ST_z(\varphi)) \quad (z \text{ a variable not in } \varphi).
\end{array}
$$

The main result in [31] is that these binders form an expressive hierarchy. If we let $<$ stand for the relation "is strictly less expressive than" then we have that $\mathcal{H}(\downarrow) < \mathcal{H}(\exists) < \mathcal{H}(\Downarrow)$ and $\mathcal{H}(\mathsf{E}) < \mathcal{H}(\Sigma) < \mathcal{H}(\Downarrow)$. The expressivity inclusions are proved using the following equivalences:

$$
\begin{array}{lcl}
\downarrow x.\varphi & \equiv & \exists x.(x \wedge \varphi) \\
\exists x.\varphi & \equiv & \Downarrow z.\Downarrow x.(z \wedge \varphi) \quad (z \text{ a variable not in } \varphi) \\
\mathsf{E}\varphi & \equiv & \Sigma z.\varphi \quad (z \text{ a variable not in } \varphi) \\
\Sigma x.\varphi & \equiv & \Downarrow z.\Downarrow x.(x \wedge \varphi) \quad (z \text{ a variable not in } \varphi).
\end{array}
$$

Moreover, the equivalence $\Downarrow x.\varphi \equiv \downarrow x.\mathsf{E}\varphi$ shows that $\mathcal{H}(\Downarrow) \leq \mathcal{H}(\downarrow, \mathsf{E})$ and hence any language containing an operator from each of the two "branches" in the hierarchy is expressively equivalent to $\mathcal{H}(\Downarrow)$. The strictness of the hierarchy is proved in [31] using different variants of bisimulations, preserving truth of formulas of the various languages. This paper also introduced so-called spypoint arguments, and used them to show that basic hybrid languages enriched with $\downarrow$ lack the finite model property and are undecidable. Spypoint arguments were later used to show a number of complexity and undecidability results in hybrid logic (see, in particular, [8]).

In [141, 142], Tzakova explores some examples of very expressive hybrid languages with binding operators in more detail, both axiomatically and by means of tableaux systems.

We turn now from motivation and historical remarks to recent developments and the current state of the field.

3 MODEL THEORY

Many different hybrid languages were introduced in the previous sections. In this section, we will discuss two languages in more detail, namely $\mathcal{H}(@)$ and $\mathcal{H}(@, \downarrow)$. These two hybrid languages have received most attention in recent literature, and the proofs of the results we will discuss can usually be adapted to other hybrid languages.

3.1 Completeness

One of the most important motivations for the study of hybrid logics has been that the addition of nominals to the modal language makes it possible to prove very general completeness results, using a straightforward adaptation of the Henkin construction for first-order logic.

DEFINITION 4. The logic $\mathbf{K}_{\mathcal{H}(@,\downarrow)}$ is the smallest set of $\mathcal{H}(@, \downarrow)$ formulas that includes all axioms, and is closed under the rules, given in Figure 2. Given a set Σ of $\mathcal{H}(@, \downarrow)$ formulas, $\mathbf{K}_{\mathcal{H}(@,\downarrow)} + \Sigma$ is the logic obtained by adding all formulas in Σ as axioms to $\mathbf{K}_{\mathcal{H}(@,\downarrow)}$, and closing again under the rules in Figure 2. Given a set of $\mathcal{H}(@)$-formulas Σ, $\mathbf{K}_{\mathcal{H}(@)}$ and $\mathbf{K}_{\mathcal{H}(@)} + \Sigma$ are defined analogous to $\mathbf{K}_{\mathcal{H}(@,\downarrow)}$ and $\mathbf{K}_{\mathcal{H}(@,\downarrow)} + \Sigma$, except without the DA axiom scheme (note that this is the only axiom or rule in which $\downarrow$ occurs).

Axioms:	
(CT)	All classical tautologies
($K_\Box$)	$\vdash [R](\varphi \to \psi) \to [R]\varphi \to [R]\psi$
($K_@$)	$\vdash @_i(\varphi \to \psi) \to @_i\varphi \to @_i\psi$
(Selfdual$_@$)	$\vdash @_i\varphi \leftrightarrow \neg @_i \neg\varphi$
(Ref$_@$)	$\vdash @_i i$
(Agree)	$\vdash @_i @_j \varphi \leftrightarrow @_j\varphi$
(Intro)	$\vdash i \to (\varphi \leftrightarrow @_i\varphi)$
(Back)	$\vdash \langle R\rangle @_i\varphi \to @_i\varphi$
(DA)	$\vdash @_i(\downarrow x.\varphi \leftrightarrow \varphi[x/i])$
Rules:	
(MP)	If $\vdash \varphi$ and $\vdash \varphi \to \psi$ then $\vdash \psi$
(Subst)	If $\vdash \varphi$ then $\vdash \varphi^\sigma$, for σ a substitution
(Gen$_@$)	If $\vdash \varphi$ then $\vdash @_i\varphi$
(Gen$_\Box$)	If $\vdash \varphi$ then $\vdash [R]\varphi$
(Name)	If $\vdash @_i\varphi$ and i does not occur in φ, then $\vdash \varphi$
(BG)	If $\vdash @_i\langle R\rangle j \to @_j\varphi$, $j \neq i$ and j does not occur in φ, then $\vdash @_i[R]\varphi$

Figure 2. Axioms and rules for $\mathbf{K}_{\mathcal{H}(@,\downarrow)}$

One note should be made, concerning the substitution rule (Subst). By this rule, one cannot only replace propositional variables uniformly by arbitrary formulas, but one can also replace nominals uniformly by other nominals (note that substituting nominals by formulas does not preserve validity in general).

We call an axiomatization *complete* with respect to a class of frames, if for all formulas φ of the relevant language, φ is derivable in the axiomatization iff φ is valid on the given frame class. An axiomatization is *strongly complete* with respect to a frame class if for every set of formulas Σ and formula φ of the relevant language, $\Sigma \models^{loc} \varphi$ iff there are $\psi_1, \ldots, \psi_n \in \Sigma$ such that $\psi_1 \wedge \cdots \wedge \psi_n \to \varphi$ is derivable.

The following completeness result is taken from [34], but slight variations of it can be found already in [37]. Recall that a formula is *pure* if it contains no propositional variables (but may possibly contain nominals).

THEOREM 5 (Pure completeness).

1. *Let Σ be any set of pure $\mathcal{H}(@)$-formulas. Then $\mathbf{K}_{\mathcal{H}(@)} + \Sigma$ is strongly complete for the class of frames defined by Σ.*

2. *Let Σ be any set of pure $\mathcal{H}(@,\downarrow)$-formulas. Then $\mathbf{K}_{\mathcal{H}(@,\downarrow)} + \Sigma$ is strongly complete for the class of frames defined by Σ.*

By *the frame class defined by* Σ, we mean the class of frames on which each formula in Σ is valid. Many frame properties can be defined using pure hybrid formulas, including properties such as irreflexivity, that cannot be defined in the basic modal language. A precise characterization of frame properties definable by pure formulas will be given in Section 3.2.

The proof of Theorem 5 trades heavily on the presence of the (Name) and (BG) rules. In [34], Blackburn and ten Cate show that, in the case of $\mathcal{H}(@,\downarrow)$, these rules (which are non-orthodox

in the sense that they involve syntactic side conditions) can be replaced by

(Name$_\downarrow$)	$\vdash \downarrow s.(s \to \varphi) \to \varphi$	provided that s does not occur in φ
(BG$_\downarrow$)	$\vdash @_i[R]\downarrow s.@_i\langle R\rangle s$	
(Gen$_\downarrow$)	If $\vdash \varphi$ then $\vdash \downarrow s.\varphi$.	

and an axiomatization with only orthodox rules is obtained, for which Theorem 5 still holds. In the case of $\mathcal{H}(@)$, on the other hand, the (Name) and (BG) rule cannot be eliminated. More precisely, every axiomatization for $\mathcal{H}(@)$ that is complete for arbitrary pure extensions contains either infinitely many rules or rules with side conditions [34].

Part of the present section will be devoted to a proof of Theorem 5. However, before we start, we will mention some other, complementary completeness results.

Theorem 5 resembles in spirit the Sahlqvist completeness theorem for modal logic (cf. Chapter 1 of this handbook). This raises the question of how pure formulas and Sahlqvist formulas relate, both in terms of expressive power and in terms of proof theoretic behaviour. As it turns out, for every modal Sahlqvist formula φ there is a pure sentence ψ of $\mathcal{H}(@,\downarrow)$ that defines the same frame class as φ, and, moreover, ψ can be picked such that $\varphi \to \psi$ is provable in $\mathbf{K}_{\mathcal{H}(@,\downarrow)}$[4]. It follows from this observation that every extension of $\mathbf{K}_{\mathcal{H}(@,\downarrow)}$ with modal Sahlqvist axioms is complete.

However, there are frame properties that can be defined by modal Sahlqvist formulas but not by pure $\mathcal{H}(@)$-formulas. For example, no set of pure $\mathcal{H}(@)$-formulas defines the same frame class as the modal Sahlqvist formula (CR) $\Diamond\Box p \to \Box\Diamond p$. This makes the following result, proved in [140], interesting.

THEOREM 6 (Sahlqvist completeness). *Let Σ be any set of modal Sahlqvist formulas. Then $\mathbf{K}_{\mathcal{H}(@)} + \Sigma$ is strongly complete for the class of frames defined by Σ.*

Completeness does not hold for arbitrary combinations of pure formulas and modal Sahlqvist formulas. Consider the Sahlqvist axiom (CR) given above and the pure formula (NoGrid) $\Diamond(i \wedge \Diamond j) \to \Box(\Diamond j \to i)$. The incompleteness of $\mathbf{K}_{\mathcal{H}(@)} + \{\text{(CR)}, \text{(NoGrid)}\}$ is proved in [140] using a general frame argument.

It should be noted that, when converse modalities are added to the language (as in the basic tense logic), modal Sahlqvist formulas *can* be translated into pure $\mathcal{H}(@)$ formulas. And, indeed, in this case axiomatizations combining pure formulas and modal Sahlqvist formulas are always strongly complete for the relevant frame class [80, 136].

There are a number of well known complete modal logics that cannot be axiomatized by means of Sahlqvist formulas, including PDL, GL and Grz. One might ask what happens when nominals and satisfaction operators are added to these logics. The following result, proved in [20, 136], provides a partial answer. It shows that, under certain condition, a complete axiomatization of a modal logic can be turned into a complete axiomatization of the corresponding hybrid logic (in the language $\mathcal{H}(@)$). Recall that a modal logic *has a master modality* if there is a modality $[*]$ that satisfies the S4 axioms, such that $[*]p \to [R]p$ is derivable for all other modalities $[R]$ in the language (see also Chapter 2 and 4 of this handbook). Furthermore, recall the notion of *admitting filtration* defined in Chapter 3 of this handbook. Informally, a logic defined over a class of frames K admits filtration if each formula φ can be associated with a set of formulas Σ_φ (the "filtration

[4] This essentially follows from the proof by substitutions of the Sahlqvist correspondence theorem (cf. Chapter 1 of this handbook), since the substitutions used only involve a bounded form of quantification. See Section 3.2 for more information on the tight relationship between bounded quantification and $\mathcal{H}(@,\downarrow)$.

set" of φ) such that for each model $\mathcal{M}$ based on a frame in K, and for each formula φ, there is a filtration of $\mathcal{M}$ over Σ_φ of which the underlying frame is in K.

THEOREM 7. *Let Σ be any set of modal formulas such that the modal logic $\mathbf{K}+\Sigma$ is complete, admits filtration and has a master modality. Then $\mathbf{K}_{\mathcal{H}(@)}+\Sigma$ is also complete.*

GL, Grz and PDL all meet the requirements of Theorem 7. Incidentally, a similar transfer result cannot exist for $\mathcal{H}(@,\downarrow)$. Indeed, the $\mathcal{H}(@,\downarrow)$-logic of the frame class defined by GL (i.e., the class of transitive and conversely well-founded frames) is not recursively axiomatizable [136].

We now prove Theorem 5 using a technique similar to that used in a standard, Henkin-style completeness proof for first-order logic [58]. The general argument runs as follows: we will show that every consistent set of formulas can be extended to a maximal consistent set satisfying certain properties. Next, we will construct out of each such maximal consistent set a model, whose domain consists of equivalence classes of nominals. Finally, we show that the constructed model satisfies the original set of formulas, and that the underlying frame satisfies the relevant frame conditions.

The proof of the following lemma is straightforward.

LEMMA 8. *The following formulas and rule are derivable in $\mathbf{K}_{\mathcal{H}(@)}+\Sigma$.*

1. $\vdash @_j k \to (@_j\psi \leftrightarrow @_k\psi)$
2. $\vdash @_j(\psi_1 \wedge \psi_2) \leftrightarrow @_j\psi_1 \wedge @_j\psi_2$
3. $\vdash @_j\neg\psi \leftrightarrow \neg @_j\psi$
4. $\vdash @_j @_k\psi \leftrightarrow @_k\psi$
5. $\vdash @_j\langle R\rangle k \wedge @_k\psi \to @_j\langle R\rangle\psi$
6. *If* $\vdash @_i\langle R\rangle j \wedge @_j\varphi \to \psi$ *then* $\vdash @_i\langle R\rangle\varphi \to \psi$, *provided* $i \neq j$ *and* j *does not occur in* φ *or* ψ.

We can now prove a Lindenbaum Lemma that shows how to extend any consistent set of formulas to a maximally consistent set, but in addition we will ensure that all diamonds are "witnessed" by nominals.

LEMMA 9. *Every $\mathbf{K}_{\mathcal{H}(@,\downarrow)}+\Sigma$-consistent set Γ can be extended to a maximal $\mathbf{K}_{\mathcal{H}(@,\downarrow)}+\Sigma$-consistent set Γ^+ such that*

1. *One of the elements of Γ^+ is a nominal;*
2. *For all variables x, there is a nominal i such that $@_i x \in \Gamma$.*
3. *For all $@_i\langle R\rangle\varphi \in \Gamma$ there is a nominal j such that $@_i\langle R\rangle j \in \Gamma$ and $@_j\varphi \in \Gamma$.*

Proof. By expanding the language with countably many nominals, we can ensure that there are infinitely many nominals that do not occur in Γ, while preserving consistency of Γ. Let $(i_n)_{n\in\mathbb{N}}$ be an enumeration of the nominals of the extended language, and let $(\varphi_n)_{n\in\mathbb{N}}$ be an enumeration of all $\mathcal{H}(@,\downarrow)$-formulas of the extended language. We will construct Γ^+ as the limit of an infinite sequence $\Gamma^0 \subseteq \Gamma^1 \subseteq \Gamma^2 \subseteq \cdots$.

Let Γ^0 denote the set $\Gamma \cup \{i\} \cup \{@_{j_x} x \mid x \in \mathsf{VAR}\}$, where the new formulas use nominals not in Γ, and the j_x are such that if x and y are different variables then also j_x and j_y are different.

It is easy to see that Γ_0 is consistent. For example, the addition of i cannot cause inconsistency because otherwise there are $\varphi_1, \ldots, \varphi_n \in \Gamma$ such that $\vdash_{\mathbf{K}_{\mathcal{H}(@,\downarrow)}+\Sigma} i \rightarrow \neg(\varphi_1 \wedge \cdots \wedge \varphi_n)$. By the (Gen$_@$) rule and the (K$_@$) axiom, it follows that $\vdash_{\mathbf{K}_{\mathcal{H}(@,\downarrow)}+\Sigma} @_i i \rightarrow @_i\neg(\varphi_1 \wedge \cdots \wedge \varphi_n)$. By the (Ref$_@$) axiom and the (MP) rule, $\vdash_{\mathbf{K}_{\mathcal{H}(@,\downarrow)}+\Sigma} @_i\neg(\varphi_1 \wedge \cdots \wedge \varphi_n)$, and hence, by the (Name) rule, $\vdash_{\mathbf{K}_{\mathcal{H}(@,\downarrow)}+\Sigma} \neg(\varphi_1 \wedge \cdots \wedge \varphi_n)$. But this contradicts the fact that Γ is consistent. The case for the additional $@_{j_x}x$ formulas is similar. Notice that the set Γ^0 satisfies already conditions 1 and 2 in the lemma. We only need to ensure condition 3.

For $k \in \mathbb{N}$, define Γ^{k+1} as follows:

1. $\Gamma^{k+1} = \Gamma^k$ if $\Gamma^k \cup \{\varphi_k\}$ is $\mathbf{K}_{\mathcal{H}(@,\downarrow)} + \Sigma$-inconsistent,
2. otherwise
   (a) $\Gamma^{k+1} = \Gamma^k \cup \{\varphi_k\}$ if φ_k is not of the form $@_i\langle R\rangle\psi$.
   (b) $\Gamma^{k+1} = \Gamma^k \cup \{\varphi_k, @_i\langle R\rangle i_m, @_{i_m}\psi\}$ if φ_k is of the form $@_i\langle R\rangle\psi$, where i_m is the first nominal that does not occur in Γ^k or φ_k.

Each step preserves consistency: if Γ^k is $\mathbf{K}_{\mathcal{H}(@,\downarrow)} + \Sigma$-consistent, then so is Γ^{k+1}. The only non-trivial case is (2.2b), and we will prove that also in this case, consistency is preserved.

Let $\Gamma^k \cup \{\varphi_k\}$ be $\mathbf{K}_{\mathcal{H}(@,\downarrow)} + \Sigma$-consistent, let φ_k be of the form $@_i\langle R\rangle\psi$, and suppose for the sake of contradiction that $\Gamma^{k+1} = \Gamma^k \cup \{\varphi_k, @_i\langle R\rangle i_m, @_{i_m}\psi\}$ is not $\mathbf{K}_{\mathcal{H}(@,\downarrow)} + \Sigma$-consistent. Then there are $\varphi_1, \ldots, \varphi_n \in \Gamma^k$ such that $\vdash_{\mathbf{K}_{\mathcal{H}(@,\downarrow)}+\Sigma} (\varphi_k \wedge @_i\langle R\rangle i_m \wedge @_{i_m}\psi) \rightarrow \neg(\varphi_1 \wedge \cdots \wedge \varphi_n)$. It follows by the last clause of Lemma 8 that $\vdash_{\mathbf{K}_{\mathcal{H}(@,\downarrow)}+\Sigma} \varphi_k \rightarrow \neg(\varphi_1 \wedge \cdots \wedge \varphi_n)$. But this contradicts the fact that $\Gamma^+ \cup \{\varphi_k\}$ is $\mathbf{K}_{\mathcal{H}(@,\downarrow)} + \Sigma$-consistent. We conclude that Γ^k is consistent.

Since $\mathbf{K}_{\mathcal{H}(@,\downarrow)} + \Sigma$-consistency is preserved at each stage, it follows that $\Gamma^+ = \bigcup_{n\in\mathbb{N}} \Gamma^n$ is $\mathbf{K}_{\mathcal{H}(@,\downarrow)} + \Sigma$-consistent. It is easy to see that Γ^+ also satisfies the other requirements in Lemma 9. ❑

We can proceed with the proof of Theorem 5.

Proof of Theorem 5. We first treat the case of $\mathcal{H}(@,\downarrow)$. Let Γ be a $\mathbf{K}_{\mathcal{H}(@,\downarrow)} + \Sigma$ consistent set of $\mathcal{H}(@,\downarrow)$-formulas and Γ^+ a maximal $\mathbf{K}_{\mathcal{H}(@,\downarrow)} + \Sigma$-consistent set of $\mathcal{H}(@,\downarrow)$-formulas extending Γ, satisfying the conditions of Lemma 9. For $i \in \mathsf{NOM}$, let $[i] = \{j \mid @_i j \in \Gamma^+\}$.

Define the hybrid model $\mathcal{M} = \langle W, (R^{\mathcal{M}})_{R\in\mathsf{REL}}, V\rangle$, where $W = \{[i] \mid i$ is a nominal occurring in $\Gamma^+\}$, $R^{\mathcal{M}} = \{([i],[j]) \mid @_i\langle R\rangle j \in \Gamma^+\}$, $V(p) = \{[i] \mid @_i p \in \Gamma^+\}$ and $V(i) = \{[i]\}$. Define the valuation g as $g(x) = [i]$ for $@_i x \in \Gamma^+$. Checking that the model and valuation that we obtain in this way are well defined is simple.

Now, for all $\mathcal{H}(@,\downarrow)$-formulas φ and nominals j, $\mathcal{M}, g, [j] \models \varphi$ iff $@_j\varphi \in \Gamma^+$. This truth lemma can be proved by a straightforward induction on φ, using the properties of Γ^+ and Lemma 8. For the inductive step for formulas of the from $\downarrow x.\psi$, we use the fact that Γ^+ contains all substitution instances of the (DA) axiom.

It follows that $\mathcal{M}, g, [i] \models \Gamma^+$, for $i \in \Gamma^+$ (recall that one of the elements of Γ^+ is a nominal). Since $\mathcal{M}$ is a named model (i.e., every point is named by a nominal) and Γ^+ contains all substitution instances of elements of Σ, all formulas in Σ are valid on the underlying frame of $\mathcal{M}$. We conclude that Γ is satisfiable on the class of frames defined by Σ.

For $\mathbf{K}_{\mathcal{H}(@)} + \Sigma$, the same argument applies. Note that the (DA) axiom was only used in the truth lemma, for the inductive step for formulas of the form $\downarrow x.\varphi$.

In the above completeness proof, the role of the non-orthodox rules (Name) and (BG) is to ensure the existence of a named model. Named models have played a crucial role in the development of the model theory of hybrid languages. As we commented in Section 2.2, they were already used by the Sofia school in their axiomatic investigations for combinatory PDL. They are closely related to the notion of a *discrete general frame*, and with the work of Venema [144] completeness for modal logics containing the difference operator D.

3.2 Expressive Power and Characterization

In this section, we investigate the expressive power of the hybrid languages $\mathcal{H}(@)$ and $\mathcal{H}(@, \downarrow)$, both on the level of models and on the level of frames, and we compare it to the basic modal language and the first-order correspondence language. For further details on the results discussed in this section see [8, 136].

Correspondence language and standard translations

From the point of view of first-order logic, nominals are nothing more than constants: they designate elements of the domain of the model. The first-order correspondence language of hybrid logic is therefore most naturally defined as follows.

DEFINITION 10. The *first-order correspondence language* for hybrid logic is the first-order language with equality that contains a unary predicate P for each propositional variable $p \in \mathsf{PROP}$, a binary relation symbol for each modality $R \in \mathsf{REL}$ and a constant for each nominal $i \in \mathsf{NOM}$.

Any hybrid model $\mathcal{M} = \langle M, (R^{\mathcal{M}})_{R \in \mathsf{REL}}, V\rangle$ can be regarded as a model for the first-order correspondence language. The accessibility relations $R^{\mathcal{M}}$ are used to interpret the binary relation symbols, unary predicates are interpreted as the subsets that V assigns to the corresponding propositional variables, and constants are interpreted as the worlds that the corresponding nominals name. In what follows, we will not distinguish between hybrid models and models for the first-order correspondence language, and we will use the notation $\mathcal{M} = \langle M, (R^{\mathcal{M}})_{R \in \mathsf{REL}}, V\rangle$ for both.

The Standard Translation from modal logic into the first-order correspondence language (cf. Chapter 1 of this handbook) can be extended to hybrid languages. The translation for the hybrid language $\mathcal{H}(\mathsf{E}, @, \downarrow)$ is given in Figure 3 (top part), where $s, t \in \mathsf{NOM} \cup \mathsf{SVAR}$, $p \in \mathsf{PROP}$, and $R \in \mathsf{REL}$. Here, we conveniently identify the state variables of hybrid logic with the variables of the first-order correspondence language.

PROPOSITION 11 (ST preserves truth). *For all hybrid formulas φ, hybrid models $\mathcal{M}$, states $w \in M$ and assignments g, $\mathcal{M}, g, m \models \varphi$ iff $\mathcal{M}, g^x_m \models ST_x(\varphi)$, where x is a variable not occurring in φ.*

As it turns out, there is also a converse translation, mapping formulas of the first-order correspondence language to formulas of $\mathcal{H}(\mathsf{E}, @, \downarrow)$. It is given in the bottom part of Figure 3.

PROPOSITION 12 (HT preserves truth). *Let φ be a formula of the first-order correspondence language. Then for every model $\mathcal{M}$, assignment g and for any state w, $\mathcal{M}, g^x_w \models \varphi$ iff $\mathcal{M}, g, w \models {\downarrow} x.HT(\varphi)$.*

It follows that $\mathcal{H}(\mathsf{E}, @, \downarrow)$ is as expressive as the first-order correspondence language. In fact, the satisfaction operators can be defined in terms of E (namely, $@_i\varphi$ is equivalent to $\mathsf{E}(i \wedge \varphi)$),

$$\begin{array}{lcl}
ST_t(\top) & = & \top \\
ST_t(s) & = & (t = s) \\
ST_t(p) & = & P(t) \\
ST_t(\neg\varphi) & = & \neg ST_t(\varphi) \\
ST_t(\varphi \wedge \psi) & = & ST_t(\varphi) \wedge ST_t(\psi) \\
ST_t(\langle R\rangle\varphi) & = & \exists y.(R(t,y) \wedge ST_y(\varphi)) \\
ST_t(\mathsf{E}\varphi) & = & \exists y.ST_y(\varphi) \\
ST_t(@_s\varphi) & = & ST_s(\varphi) \\
ST_t(\downarrow z.\varphi) & = & \exists z.(z = t \wedge ST_t(\varphi))
\end{array}$$

where y is a variable distinct from the term t and not occurring in φ

$$\begin{array}{lcl}
HT(\top) & = & \top \\
HT(R(s,s')) & = & @_s\langle R\rangle s' \\
HT(P(s)) & = & @_s p \\
HT(s = t) & = & @_s t \\
HT(\neg\varphi) & = & \neg HT(\varphi) \\
HT(\varphi \wedge \psi) & = & HT(\varphi) \wedge HT(\psi) \\
HT(\exists x.\varphi) & = & \mathsf{E}\!\downarrow\! x.HT(\varphi)
\end{array}$$

Figure 3. Standard Translation ST and Hybrid Translation HT

and therefore $\mathcal{H}(\mathsf{E}, \downarrow)$ is already as expressive as the first-order correspondence language[5]. This leaves the question open of what is the range of ST for languages weaker than $\mathcal{H}(\mathsf{E}, @, \downarrow)$, i.e., which formulas of the first-order correspondence language are (equivalent to) translations of formulas of these hybrid languages? We will discuss this issue in the next section.

Characterizing expressivity on models

In this section, we address in detail the question of which formulas of the first-order correspondence language are equivalent to (standard translations of) hybrid formulas.

First, let us generalize the notion of bisimulation to hybrid languages.

DEFINITION 13. Let $\mathcal{M} = \langle M, (R^{\mathcal{M}})_{R\in\mathsf{REL}}, V\rangle$ and $\mathcal{N} = \langle N, (S^{\mathcal{N}})_{S\in\mathsf{REL}}, U\rangle$ be hybrid models. A *hybrid bisimulation between* $\mathcal{M}$ *and* $\mathcal{N}$ is a non-empty binary relation $Z \subseteq M \times N$ such that the following clauses hold

(atom) If $Z(m,n)$, then $m \in V(p)$ iff $n \in U(p)$, for $p \in \mathsf{PROP} \cup \mathsf{NOM}$.

(nom) If $V(i) = \{m\}$ and $U(i) = \{n\}$ then $Z(m,n)$, for $i \in \mathsf{NOM}$.

(forth) If $Z(m,n)$ and $R^{\mathcal{M}}(m,m')$, then there is an $n' \in N$ such that $S^{\mathcal{N}}(n,n')$ and $Z(m',n')$.

(back) If $Z(m,n)$ and $S^{\mathcal{N}}(n,n')$, then there is an $m' \in M$ such that $R^{\mathcal{M}}(m,m')$ and $Z(m',n')$.

A formula $\varphi(x_1,\ldots,x_n)$ of the first-order correspondence language is said to be *invariant for bisimulations* if for all bisimulations Z between hybrid models $\mathcal{M}$ and $\mathcal{N}$ and for all assignments g and h with $Z(g(x_k), h(x_k))$ for $k = 1\ldots n$, it is the case that $\mathcal{M}, g \models \varphi$ iff $\mathcal{N}, h \models \varphi$.

[5] A similar translation can be given for $\mathcal{H}(@, \forall)$, see [32].

THEOREM 14. *A formula φ of the first-order correspondence language with at most one free variable x is equivalent to the standard translation of an $\mathcal{H}(@)$-formula iff φ is invariant under hybrid bisimulations.*

The proof is a straightforward generalization of the one for the basic modal language. As a corollary of Theorem 14, we obtain the following syntactic characterization.

COROLLARY 15. *A formula φ of the first-order correspondence language with at most one free variable x is equivalent to the standard translation of an $\mathcal{H}(@)$-formula iff φ is equivalent to a formula generated by the following recursive definition, where t is a term (constant or variable), c is a constant, and x is a variable distinct from t:*

$$\varphi ::= \top \mid P(t) \mid t = c \mid \neg\varphi \mid \varphi \wedge \psi \mid \exists x.(R(t,x) \wedge \varphi).$$

Proof. One direction of the claim follows from the fact that $ST(\varphi)$ is of the given form, for each $\mathcal{H}(@)$-formula φ. As for the other direction, a straightforward induction shows that every first-order formula of the given form is invariant under hybrid bisimulations, and hence every such formula with at most one free variable is equivalent to (the standard translation of) an $\mathcal{H}(@)$-formula. ❑

Let us now consider the language $\mathcal{H}(@, \downarrow)$. First, we will give a syntactic characterization (see [8] for further details). Call a first-order formula *bounded* if it is built up from atomic formulas using the Boolean connectives and bounded quantification of the form $\exists x.(R(s,x) \wedge \cdot)$ or $\forall x.(R(s,x) \rightarrow \cdot)$, where s is a term distinct from the variable x.

THEOREM 16. *A formula φ of the first-order correspondence language with one free variable is equivalent to the standard translation of a $\mathcal{H}(@, \downarrow)$ sentence iff φ is equivalent to a bounded formula.*

Proof. The standard translation of an $\mathcal{H}(@, \downarrow)$ sentence is always a bounded formula of the correspondence language. Conversely, we can extend the translation HT given in Figure 3 with the following clause for bounded quantification:

$$HT(\exists x.(R(s,x) \wedge \psi)) \;=\; @_s\langle R\rangle{\downarrow}x.HT(\psi).$$

In this way, we obtain, for each bounded formula φ of the first-order correspondence language, an $\mathcal{H}(@, \downarrow)$-formula $HT(\varphi)$. Moreover, a straightforward inductive argument shows that $HT(\varphi)$ is equivalent to φ, in the sense of Proposition 12. Recall that the formula φ in the statement of the Theorem contains at most one free variable x, and let φ' be any bounded formula equivalent to φ. It follows that φ' (and hence φ) is equivalent to $ST_x({\downarrow}x.HT(\varphi'))$. ❑

In other words, $\mathcal{H}(@, \downarrow)$ corresponds to the bounded fragment of first-order logic. By means of the notion of generated submodels, we can semantically characterize this fragment.

DEFINITION 17. Let $\mathcal{M} = \langle M, (R^{\mathcal{M}})_{R\in\mathsf{REL}}, V\rangle$ and $\mathcal{N} = \langle N, (R^{\mathcal{N}})_{R\in\mathsf{REL}}, V'\rangle$ be hybrid models. Then $\mathcal{N}$ is a *generated submodel* of $\mathcal{M}$ if $N \subseteq M$ and for all $w, v \in M$ and for any relation R_i, if $w \in N$ and $R_i(w,v)$ then $v \in N$, while R'_i and V' are the restrictions of R_i and V to N respectively. A formula φ is *invariant for generated submodels* if for all models $\mathcal{M}$, $\mathcal{N}$ such that $\mathcal{N}$ is a generated submodel of $\mathcal{M}$, and for all $\mathcal{N}$-assignments g, $\mathcal{M}, g \models \varphi$ if and only if $\mathcal{N}, g \models \varphi$.

THEOREM 18. *A formula φ of the first-order correspondence language is invariant under generated submodels iff φ is equivalent to a bounded formula.*

Proof. Suppose a first-order formula φ is invariant under generated submodels. For convenience, we assume that φ is a sentence (free variables can be replaced by new constants). Let $c_1, \ldots, c_k$ be the constants and $R_1, \ldots, R_m$ be the binary relations occurring in φ, and let P be a new unary predicate. We will use $R(s,t)$ as a shorthand for $\bigvee_{1 \leq i \leq m} R_i(s,t)$. Then the following holds:

$$\{\forall x.(R^n(c_l, x) \to P(x)) \mid 1 \leq l \leq k \text{ and } n \in \mathbb{N}\} \models \varphi \leftrightarrow \varphi^P,$$

where $R^n(x,y)$ is a shorthand for a bounded formula which expresses that y can be reached from x in exactly n steps along R (i.e., $\exists x_1(R(x,x_1) \wedge \exists x_2(R(x_1,x_2) \wedge \cdots(\cdots \wedge x_n = y)\cdots)))$ and φ^P is the result of relativising all quantifiers in φ by P (that is, $\exists x.\varphi$ becomes $\exists x.(P(x) \wedge \varphi)$ and $\forall x.\varphi$ becomes $\forall x.(P(x) \to \varphi)$). By compactness, it follows that there is an $m \in \mathbb{N}$ such that

$$\forall x.(\Big(\bigvee_{1 \leq l \leq k} R^{\leq m}(c_l, x)\Big) \to P(x)) \models \varphi \leftrightarrow \varphi^P.$$

Let φ' be the result of relativising all quantifiers in φ by the formula $\big(\bigvee_{1 \leq l \leq k}(R^{\leq m}(c_l,x))\big)$. It follows that $\models \varphi \leftrightarrow \varphi'$. Finally, φ' is (modulo some simple syntactic manipulations) a bounded sentence. ❑

This result was first proved in the sixties by Feferman and Kreisel [60, 59], and was independently proved by Areces, Blackburn and Marx [8] in the context of hybrid logic.

For any model $\mathcal{M}$ and world w, let $\mathcal{M}_w$ denote the smallest generated submodel of $\mathcal{M}$ containing w. In fact, it is easy to see that the domain of $\mathcal{M}$ contains precisely those worlds that are reachable in finitely many steps from w or from a world named by a nominal. As a corollary of the above results, we know that $\mathcal{M}, w$ and $\mathcal{M}_w, w$ agree on all sentences of $\mathcal{H}(@, \downarrow)$. If we combine this with the fact that all first-order formulas are invariant under potential isomorphisms, we obtain the following:

PROPOSITION 19. *Let $\mathcal{M}$ and $\mathcal{N}$ be models, with corresponding states w, v. If there is a potential isomorphism between $\mathcal{M}_w$ and $\mathcal{N}_v$ connecting w to v, then $\mathcal{M}, w$ and $\mathcal{N}, v$ agree on all $\mathcal{H}(@, \downarrow)$-sentences.*

While the converse does not hold in general, it does hold on ω-saturated models. This means that "potential isomorphisms between point-generated submodels" capture $\mathcal{H}(@, \downarrow)$-indistinguishability in exactly the same way that potential isomorphisms capture first-order indistinguishability.

Characterizing frame definability

Given a set of hybrid formulas Σ, we say that the *frame class defined by* Σ is the class of frames in which every formula of Σ is valid. We say that a frame class is *elementary* (or *first-order definable*) if it is defined by a first-order sentence, in the language with equality and a relation symbol for each $R \in \mathsf{REL}$. The Goldblatt-Thomasson theorem tells us that an elementary frame class is definable by a set of formulas of the basic modal language iff the class is closed under disjoint unions, generated subframes, and bounded morphic images, and its complement is closed under ultrafilter extensions (see Chapter 5 of this handbook for this result and for a definition of the notions involved). In this section, we discuss analogues of this result for hybrid languages.

Due to the increased expressivity of hybrid languages, frame classes definable by hybrid formulas are in general not closed under disjoint unions or bounded morphic images. For example, the class of irreflexive frames, which is not closed under bounded morphic images, is defined in $\mathcal{H}(@)$ by the formula $i \to \neg\Diamond i$, and the class of frames that have exactly one element, which

is not closed under disjoint unions, is defined by the formula i. Nevertheless, frame classes definable in $\mathcal{H}(@)$ are closed under generated subframes, and their complement is closed under ultrafilter extensions. In fact, a slightly stronger closure condition holds, involving a restricted form of bounded morphisms.

DEFINITION 20. Let $\mathcal{F}$ and $\mathcal{G}$ be frames, and let $\mathfrak{ue}\mathcal{G}$ be an ultrafilter extension of $\mathcal{G}$. $\mathcal{G}$ is an *ultrafilter morphic image* of $\mathcal{F}$ if there is a surjective bounded morphism $f : \mathcal{F} \to \mathfrak{ue}\mathcal{G}$ such that $|f^{-1}(u)| = 1$ for all principal ultrafilters $u \in \mathfrak{ue}\mathcal{G}$.

Note first that whenever $\mathcal{G}$ is an ultrafilter morphic image of a frame $\mathcal{F}$, $\mathfrak{ue}\mathcal{G}$ is a bounded morphic image of $\mathcal{F}$. It follows that the validity of modal formulas is preserved under taking ultrafilter morphic images. Secondly, note that every frame is an ultrafilter morphic image of its ultrafilter extension. Hence, if a property of frames is preserved under ultrafilter morphic images, its complement is preserved under taking ultrafilter extensions.

PROPOSITION 21. *All frame classes definable by a set of $\mathcal{H}(@)$-formulas are closed under taking ultrafilter morphic images.*

Proof. Let φ be an $\mathcal{H}(@)$-formula, let $f : \mathcal{F} \to \mathfrak{ue}\mathcal{G}$ be a surjective ultrafilter morphism, and suppose $\mathcal{G} \not\models \varphi$. We will show that $\mathcal{F} \not\models \varphi$.

Let V be a valuation and w a world such that $\langle \mathcal{G}, V\rangle, w \not\models \varphi$. Define the valuation $V^{\mathfrak{ue}}$ on $\mathfrak{ue}\mathcal{G}$ such that $V^{\mathfrak{ue}}(p) = \{u \mid V(p) \in u\}$ for all propositional variables p and $V^{\mathfrak{ue}}(i) = \{u \mid V(i) \in u\}$ for all nominals i. It is easily seen that $V^{\mathfrak{ue}}$ assigns to each nominal a singleton set consisting of a principal ultrafilter, and hence $V^{\mathfrak{ue}}$ is a well-defined hybrid valuation. Moreover, a standard argument [28, Proposition 2.59] shows that for all worlds v and formulas ψ, $\langle \mathcal{G}, V\rangle, v \models \psi$ iff $\langle \mathfrak{ue}\mathcal{G}, V^{\mathfrak{ue}}\rangle, \pi v \models \psi$, where πv is the principal ultrafilter generated by v. It follows that $\langle \mathfrak{ue}\mathcal{G}, V^{\mathfrak{ue}}\rangle, \pi w \not\models \varphi$.

Next, define the valuation V' for $\mathcal{F}$ such that $V'(p) = \{v \mid f(v) \in V^{\mathfrak{ue}}(p)\}$ for all propositional variables p and $V'(p) = \{v \mid f(v) \in V^{\mathfrak{ue}}(i)\}$ for all nominals i. Since f is injective on principal ultrafilters and nominals denote principal ultrafilters in $\mathfrak{ue}\mathcal{G}$, $V'(i)$ is a singleton for all nominals i, and hence $\langle \mathcal{F}, V'\rangle$ is a well-defined hybrid model. Furthermore, a standard argument shows that (the graph of) f is a hybrid bisimulation between $\mathfrak{ue}\mathcal{G}$ and $\mathcal{F}$. Since f is surjective, there is a $u \in \mathcal{F}$ such that $f(u) = \pi w$. By invariance under hybrid bisimulations, $\langle \mathcal{F}, V'\rangle, u \not\models \varphi$, and hence $\mathcal{F} \not\models \varphi$ ❑

We can strengthen Proposition 21 to the following characterization of frame definability in $\mathcal{H}(@)$ [136].

THEOREM 22. *An elementary class of frames is definable by a set of $\mathcal{H}(@)$ formulas iff it is closed under taking ultrafilter morphic images and generated subframes.*

Proof. The easy direction is already discussed above: every frame class defined by a set of $\mathcal{H}(@)$-formulas is closed under taking ultrafilter morphic images and generated subframes. We will now prove the hard direction. Let K be any elementary frame class closed under taking ultrafilter morphic images and generated subframes, and let $Th(K)$ be the set of $\mathcal{H}(@)$-formulas valid on K. To show that K is $\mathcal{H}(@)$-definable, it suffices to show that $Th(K)$ itself defines K.

Suppose that $\mathcal{F} \models Th(K)$ for some frame $\mathcal{F}$ with domain W. For each subset $A \subseteq W$, introduce a propositional variable p_A, and for each $w \in W$, introduce a nominal i_w[6]. Let Δ be

[6]Technically, this might involve adding uncountably many propositional variables and nominals to the language. However, this will not cause any problems below. Of course, individual formulas can only contain finitely many propositional variables and nominals.

the set consisting of the following formulas, for all $A \subseteq W$, $v \in W$ and $R \in \mathsf{REL}$.

$$\begin{array}{ll} p_{-A} \leftrightarrow \neg p_A & \\ p_{A\cap B} \leftrightarrow p_A \wedge p_B & \\ p_{R^{-1}(A)} \leftrightarrow \langle R \rangle p_A & \text{where } R^{-1}(A) = \{w \in W \mid \exists v \in A \text{ such that } wRv\} \\ i_v \leftrightarrow p_{\{v\}}. & \end{array}$$

Let $\Delta_{\mathcal{F}} = \{@_{i_v}[R_{i_1}]\cdots[R_{i_n}]\delta \mid v \in W, \delta \in \Delta, \text{ and } R_{i_1}, \ldots, R_{i_n} \in \mathsf{REL} \text{ with } n \in \mathbb{N}\}$. Intuitively, $\Delta_{\mathcal{F}}$ provides a full description of the frame $\mathcal{F}$. Clearly, $\Delta_{\mathcal{F}}$ is satisfiable on $\mathcal{F}$ under the natural valuation that sends p_A to A and i_v to $\{v\}$. We claim that $\Delta_{\mathcal{F}}$ is satisfiable on K. By compactness (recall that K is elementary), it suffices to show that every finite conjunction δ of elements of $\Delta_{\mathcal{F}}$ is satisfiable on K. But this follows immediately: δ is satisfiable on $\mathcal{F}$ and $\mathcal{F} \models \mathit{Th}(K)$, hence $\neg\delta \notin \mathit{Th}(K)$, i.e., δ is satisfiable on K.

Let $\langle \mathcal{G}, V \rangle \models \Delta_{\mathcal{F}}$ with $\mathcal{G} \in K$. Since K is closed under generated subframes, we may assume that $\mathcal{G}$ is generated by the set of points that are named by a nominal. It then follows that the model $\langle \mathcal{G}, V \rangle$ globally satisfies Δ. Let $\langle \mathcal{G}^*, V^* \rangle$ be an ω-saturated elementary extension of $\langle \mathcal{G}, V \rangle$ (such elementary extensions are known to exist even in the case of uncountable vocabularies). By elementarity, $\mathcal{G}^* \in K$ and $\langle \mathcal{G}^*, V^* \rangle$ globally satisfies Δ.

It can be shown that $\mathfrak{ue}\mathcal{F}$ is an ultrafilter morphic image of $\mathcal{G}^*$, where the ultrafilter morphism f is given by $f(v) = \{A \subseteq W \mid \langle \mathcal{G}^*, V^* \rangle, v \models p_A\}$. See [136] for further details. Since K is closed under ultrafilter morphic images, we conclude that $\mathcal{F} \in K$. ❑

As we already discussed earlier, there is a particular interest in frame conditions definable by *pure* formulas, since these immediately yield complete axiomatizations. It would be worth having a characterization of the properties of frames that can be defined using pure formulas only. Details for such results can be found in [136], here we only state one theorem.

DEFINITION 23. We say that a bisimulation Z between frames $\mathcal{F} = \langle F, (R^{\mathcal{F}})_{R\in\mathsf{REL}} \rangle$ and $\mathcal{G} = \langle G, (R^{\mathcal{G}})_{R\in\mathsf{REL}} \rangle$ *respects a set X of elements* of $\mathcal{G}$ if for all $x \in X$,

1. $Z(w, x)$ and $Z(v, x)$ implies $w = v$, and

2. $Z(w, x)$ and $Z(w, v)$ implies $v = x$.

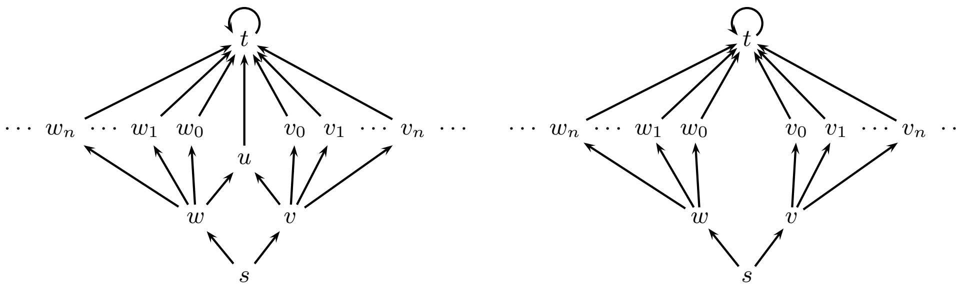
A *bisimulation system* from $\mathcal{F}$ to $\mathcal{G}$ is a function f that assigns to each finite subset $X \subseteq G$ a total bisimulation $f(X) \subseteq F \times G$ respecting X.

THEOREM 24. *A class of frames is defined by a pure $\mathcal{H}(@)$-formula iff it is elementary and closed under taking images of bisimulation systems.*

An example of a frame condition that is not preserved under taking images of bisimulation systems is the Church-Rosser property.

PROPOSITION 25. *The frame condition $\forall xyz.(R(x, y) \wedge R(x, z) \rightarrow \exists u.(R(y, u) \wedge R(z, u)))$ is not preserved under images of bisimulation systems.*

Proof. Consider the two frames $\mathcal{F}_1 = \langle F_1, R^{\mathcal{F}_1} \rangle$ and $\mathcal{F}_2 = \langle F_2, R^{\mathcal{F}_2} \rangle$ shown in Figure 4. Notice that $\mathcal{F}_1$ is identical to $\mathcal{F}_2$, except for the additional point u (and its incoming and outgoing arrows). For any finite set $X \subseteq F_2$, let $f(X) = \{(w, w) \mid w \in F_1\} \cup \{(u, w_k), (u, v_l)\}$, for some $w_k, v_l \notin X$ (note that such w_k and v_l always exist). As is not hard to see, f is a bisimulation system. However, $\mathcal{F}_1$ satisfies the frame condition, while $\mathcal{F}_2$ does not. ❑



$\mathcal{F}_1$ $\mathcal{F}_2$

Figure 4. Church-Rosser is not definable by pure formulas

It follows that the Church-Rosser property cannot be defined by pure formulas of $\mathcal{H}(@)$. A similar example of indefinability is the class of transitive and atomic frames (where *atomicity* means that $\forall x.\exists y.(R(x,y) \wedge \forall z.(R(y,z) \to z = y)))$. This class of frames is defined by the modal formula $(\Diamond\Diamond p \to \Diamond p) \wedge (\Box\Diamond p \to \Diamond\Box p)$, but it cannot be defined by means of pure $\mathcal{H}(@)$-formulas, since it is not closed under images of bisimulation systems.

Finally, let us consider the language $\mathcal{H}(@,\downarrow)$. Interestingly, here the difference in frame definable power between pure formulas and arbitrary formulas is much smaller. In fact, every elementary frame property that can be defined by a set of $\mathcal{H}(@,\downarrow)$-sentences can already be defined by means of a single pure $\mathcal{H}(@,\downarrow)$-sentence. A precise characterization is given in the following theorem.

DEFINITION 26. A frame $\mathcal{F}$ is a *finitely generated subframe* of a frame $\mathcal{G}$, if there is a finite set X of elements of the domain of $\mathcal{G}$, such that $\mathcal{F}$ is the submodel of $\mathcal{G}$ generated by X (i.e., such that $\mathcal{F}$ is the smallest generated submodel of $\mathcal{G}$ whose domain contains all elements of X).

We say that a frame class K *reflects finitely generated subframes* whenever it is the case for all frames $\mathcal{F}$ that, if every finitely generated subframe of $\mathcal{F}$ is in K, then $\mathcal{F} \in K$.

THEOREM 27. *Let K be an elementary frame class. Then the following are equivalent:*

1. *K is defined by a set of $\mathcal{H}(@,\downarrow)$ sentences.*
2. *K is defined by a single pure $\mathcal{H}(@,\downarrow)$ sentence.*
3. *K is closed under taking generated subframes, and reflects finitely generated subframes.*

This result can be extended to formulas containing only a limited number of nominals: let us say that a frame class K reflects n-point generated subframes whenever it is the case for all frames $\mathcal{F}$ that, if every subframe of $\mathcal{F}$ generated by at most n elements is in K, then $\mathcal{F} \in K$. Then Theorem 27 can be refined to the following result [8, 136].

THEOREM 28. *Let K be an elementary frame class and $n \in \mathbb{N}$. Then the following are equivalent:*

1. *K is defined by a set of $\mathcal{H}(@,\downarrow)$ sentences containing (all together) at most n nominals.*
2. *K is defined by a single pure $\mathcal{H}(@,\downarrow)$ sentence containing at most n nominals.*

Language	*Frame classes defined by arbitrary formulas*	*Frame classes defined by pure formulas*
$\mathcal{H}$	closed under ultrafilter morphic images, and if every point-generated subframe of a frame $\mathcal{F}$ is a proper generated subframe of a frame in the class, then $\mathcal{F}$ is in the class	closed under images of bisimulation systems, and if every point-generated subframe of a frame $\mathcal{F}$ is a proper generated subframe of a frame in the class, then $\mathcal{F}$ is in the class
$\mathcal{H}(@)$	closed under ultrafilter morphic images and generated subframes	closed under images of bisimulation systems and generated subframes
$\mathcal{H}(\mathsf{E})$	closed under ultrafilter morphic images	closed under images of bisimulation systems
$\mathcal{H}(@, \downarrow)$	closed under generated subframes and reflecting finitely generated subframes.	closed under generated subframes and reflecting finitely generated subframes.

Figure 5. Elementary frame classes definable in $\mathcal{H}$, $\mathcal{H}(@)$, $\mathcal{H}(\mathsf{E})$ and $\mathcal{H}(@, \downarrow)$

3. *K is closed under taking generated subframes, and reflects $(n + 1)$-generated subframes.*

Note that every modally definable frame class is closed under generated subframes and reflects point-generated subframes. It follows by the above result that every *elementary* modally definable frame class (in particular, every frame class defined by a modal Sahlqvist formula), is defined by a nominal-free pure sentence of $\mathcal{H}(@, \downarrow)$.

The most important results of this section are summarized in Figure 5 which also contains analogous results for the languages $\mathcal{H}$ and $\mathcal{H}(\mathsf{E})$. Again, full details can be found in [136].

3.3 *Interpolation and Beth Definability*

We will now turn to the properties of interpolation and Beth definability. The results in this section are mainly based on [8, 136].

Recall that the modal logic of a class of frames K has interpolation if whenever $\varphi \to \psi$ is valid in K, then there exists a formula θ (called the *interpolant*) such that $\varphi \to \theta$ and $\theta \to \psi$ are valid in K, and all propositional variables occurring in θ occur both in φ and in ψ[7]. This definition can be generalized to hybrid logics in two ways, depending on whether only the propositional variables or also the nominals occurring in the interpolant are required to occur both in φ and in ψ. We will say that a hybrid logic has interpolation *over propositional variables* or *over propositional variables and nominals* to distinguish between these definitions.

The basic hybrid language $\mathcal{H}(@)$ lacks interpolation over propositional variables and nominals [8], as can be seen by the valid implication $i \wedge \Diamond i \to (j \to \Diamond j)$. An interpolant to this implication (which should express that the actual world is related to itself) is not allowed to contain any nominals. It is easily seen, using a bisimulation argument, that no such interpolant exists. Interpolation over propositional variables *does* hold. In fact, it holds relative to many frame classes [140, 136]:

THEOREM 29. *$\mathcal{H}(@)$ has interpolation over propositional variables relative to any frame class definable by a set of first-order universal Horn sentences.*

[7]This is sometimes called *local interpolation* or *arrow interpolation*, and in particular we are presenting it in its semantic version. We will not discuss global interpolation.

For $\mathcal{H}(@, \downarrow)$, we have better results: it has interpolation over propositional variables and nominals relative to many frame classes [8, 136]:

THEOREM 30. *$\mathcal{H}(@, \downarrow)$ has interpolation over propositional variables and nominals relative to any frame class definable by a set of nominal-free $\mathcal{H}(@, \downarrow)$ sentences. Moreover, $\mathcal{H}(@, \downarrow)$ has interpolation over propositional variables relative to any frame class definable by a set of $\mathcal{H}(@, \downarrow)$-sentences (possibly containing nominals).*

Theorem 30 covers many frame classes. Indeed, we saw in the previous section that every modally definable elementary frame class can be defined by a nominal-free sentence of $\mathcal{H}(@, \downarrow)$. It was shown in [30] that the interpolants can be effectively computed from a tableau proof (see also Section 5.3)[8]. The interpolation algorithm presented in [30] is conservative: on purely modal input it computes interpolants in which the hybrid syntactic machinery does not occur.

Given that $\mathcal{H}(@)$ lacks interpolation over propositional variables and nominals and $\mathcal{H}(@, \downarrow)$ has it, and given that $\mathcal{H}(@, \downarrow)$ has an undecidable satisfiability problem (as we will see in the next section), it is natural to ask whether there is any decidable hybrid language with interpolation over propositional variables and nominals. The answer is negative [135]: every extension of the minimal hybrid language $\mathcal{H}$ (satisfying certain regularity conditions such as allowing substitution of one nominal by another) either lacks interpolation or is undecidable. Moreover, $\mathcal{H}(@, \downarrow)$ is the least expressive extension of $\mathcal{H}(@)$ (satisfying the same regularity conditions) with interpolation over propositional variables and nominals.

The following can be seen as a weak version of this result. The proof is illustrative.

THEOREM 31. *If $\mathcal{H}(@)$ has interpolation over propositional variables and nominals on a frame class K, then $\mathcal{H}(@)$ is expressively complete for $\mathcal{H}(@, \downarrow)$ on K (i.e., for each formula $\varphi \in \mathcal{H}(@, \downarrow)$, there exist a formula $\varphi' \in \mathcal{H}(@)$ such that φ and φ' are equivalent on K).*

Proof. Assume that $\mathcal{H}(@)$ has interpolation over propositional variables and nominals on K. We will show that every $\mathcal{H}(@, \downarrow)$ sentence φ is equivalent (on K) to an $\mathcal{H}(@)$-formula, proceeding by induction on the length of φ. The only interesting case here is when φ is of the form $\downarrow x.\psi(x)$. Let i and j be nominals not occurring in $\downarrow x.\psi(x)$. By induction, we know that $\psi(i)$ and $\psi(j)$ are equivalent to $\mathcal{H}(@)$-formulas $\psi'(i)$ and $\psi'(j)$ respectively. Now, the following implication is valid:

$$K \models i \wedge \psi'(i) \to (j \to \psi'(j)).$$

Any interpolant θ for this implication is equivalent to $\downarrow x.\psi(x)$. For, consider any model $\mathcal{M}$ and world w such that $\mathcal{M}, w \models \downarrow x.\psi(x)$. Let $\mathcal{M}[i/w]$ be the model that differs from $\mathcal{M}$ only in the fact that w is the denotation of i. Since i does not occur in $\downarrow x.\psi(x)$, we have that $\mathcal{M}[i/w], w \models \downarrow x.\psi(x)$, hence $\mathcal{M}[i/w], w \models i \wedge \psi(i)$. It follows that $\mathcal{M}[i/w], w \models \theta$. Since i does not occur in θ, it follows that $\mathcal{M}, w \models \theta$. Conversely, suppose $\mathcal{M}, w \models \theta$. Let $\mathcal{M}[j/w]$ be the model that differs from $\mathcal{M}$ only in the fact that j denotes w. Since j does not occur in θ, we have that $\mathcal{M}[j/w], w \models \theta$. It follows that $\mathcal{M}[j/w], w \models j \to \psi(j)$, and hence $\mathcal{M}[j/w], w \models \downarrow x.\psi(x)$. Since j does not occur in $\downarrow x.\psi(x)$, it follows that $\mathcal{M}, w \models \downarrow x.\psi(x)$. ❑

To conclude our discussion on interpolation, we consider the notion of *uniform interpolants*. As is discussed in Chapter 8 of this handbook, the modal logics K, S5, Grz and GL enjoy a very special form of interpolation, called uniform interpolation. For any formula φ, let $\mathsf{PROP}(\varphi)$ be the set of propositional variables occurring in φ. Then a modal logic has uniform interpolation

[8] Theorem 30 is related to a result by Feferman and Kreisel [60, 59] who proved that the bounded fragment of first-order logic has interpolation by means of a cut free sequent calculus.

if for every formula φ and for any $P \subseteq \mathsf{PROP}(\varphi)$, there is a formula φ_P (called a *uniform interpolant*) such that for any formula ψ, if $\mathsf{PROP}(\psi) \cap \mathsf{PROP}(\varphi) \subseteq P$ and $\varphi \to \psi$ is derivable, then $\varphi_P \to \psi$ is derivable.

We can generalize the definition to hybrid logics, and say that a hybrid logic has *uniform interpolation over propositional variables* if for every formula φ and for any $P \subseteq \mathsf{PROP}(\varphi)$, there is a formula φ_P such that for any formula ψ, if $\mathsf{PROP}(\psi) \cap \mathsf{PROP}(\varphi) \subseteq P$, and all nominals occurring in ψ occur in φ, then $\varphi \to \psi$ is valid iff $\varphi_P \to \psi$ is valid. Note the requirement imposed on nominals in this definition. It turns out that the $\mathcal{H}(@)$-logics of the frame classes corresponding to the modal logics K, S5, Grz and GL have uniform interpolation over propositional variables [20, 136].

Finally, to close this section we turn to the Beth definability property. Recall that a logic is said to have the Beth Definability property if, intuitively, every implicit definition can be made explicit. More precisely, let $\Gamma(p)$ be any set of formulas containing the propositional variables p and possibly other propositional variables and nominals. $\Gamma(p)$ defines p implicitly if in all models in which both $\Gamma(p)$ and $\Gamma(p')$ are true at every state, also $p \leftrightarrow p'$ is true at every state (here, p' is a propositional variable not occurring in $\Gamma(p)$, and $\Gamma(p')$ is obtained from $\Gamma(p)$ by replacing all occurrences of p by p'). In other words, $\Gamma(p)$ defines p implicitly if $\Gamma(p) \cup \Gamma(p') \models^{glo} p \leftrightarrow p'$, where $\models^{glo}$ denotes global entailment. The Beth property states that whenever $\Gamma(p)$ defines p implicitly, there exists a formula θ in which p does not occur, such that $\Gamma(p) \models^{glo} p \leftrightarrow \theta$[9]. Clearly, θ is an explicit definition of p, relative to the theory $\Gamma(p)$.

The Beth definability property for a logic is typically established as a corollary of the interpolation property for propositional variables. In particular, the following theorem can be shown using the above interpolation results.

THEOREM 32. *$\mathcal{H}(@, \downarrow)$ has the Beth definability property relative to any frame class defined by a set of $\mathcal{H}(@, \downarrow)$ sentences, and $\mathcal{H}(@)$ has the Beth definability property relative to any frame class defined by a set of first-order universal Horn formulas.*

Surprisingly, the minimal hybrid language $\mathcal{H}$ lacks the Beth property relative to the class of all frames [20].

4 DECIDABILITY AND COMPLEXITY

In this section, we will review the complexity of the satisfiability problem for various hybrid logics. First, let us consider the language $\mathcal{H}(@)$. We start with some good news: the satisfiability problem of $\mathcal{H}(@)$ is PSPACE-complete [6]. We provide a game based argument for the upper bound.

THEOREM 33. *$\mathcal{H}(@)$-satisfiability on the class of all frames is* PSPACE*-complete.*

Proof. We only discuss the mono-modal case (the multi-modal case is a simple extension). The lower bound follows from the PSPACE-hardness of classical modal logic. We show the upper bound by defining, given a formula φ, the notion of a φ-game between two players. We will show that the existential player has a winning strategy for the φ-game iff φ is satisfiable. Moreover, every φ-game stops after at most as many rounds as the modal depth of φ and the information on the playing board is polynomial in the length of φ. This implies that a PSPACE algorithm exists. Fix a formula φ and let d be the number of different nominals appearing in

[9] This is sometimes called the *global Beth property*. We will not discuss the local Beth property here.

φ. A φ-Hintikka set is a maximal consistent set of subformulas of φ. We denote the set of subformulas of φ by $\mathsf{SF}(\varphi)$. The φ-game is played as follows. There are two players, $\forall$belard (male) and $\exists$loise (female). She starts the game by playing a collection $\{X_0, \dots, X_l\}, l \leq d$ of Hintikka sets and specifying a relation R on them. $\exists$loise loses immediately if one of the following conditions is false:

1. X_0 contains φ, and all others X_l contain at least one nominal occurring in φ.
2. no nominal occurs in two different Hintikka sets.
3. for all X_l, for all $@_i\psi \in \mathsf{SF}(\varphi)$, $@_i\psi \in X_l$ iff $\{i, \psi\} \subseteq X_k$, for some k.
4. for all $\Diamond\psi \in \mathsf{SF}(\varphi)$, if $R(X_l, X_k)$ and $\Diamond\psi \notin X_l$, then $\psi \notin X_k$.

Now $\forall$belard may choose an X_l and a "defect-formula" $\Diamond\psi \in X_l$. $\exists$loise must respond with a Hintikka set Y such that

1. $\psi \in Y$ and for all $\Diamond\theta \in \mathsf{SF}(\varphi)$, $\Diamond\theta \notin X_l$ implies that $\theta \notin Y$.
2. for all $@_i\psi \in \mathsf{SF}(\varphi)$, $@_i\psi \in Y$ iff $\{i, \psi\} \subseteq X_k$, for some k.
3. if $i \in Y$ for some nominal i, then Y is one of the Hintikka sets she played at the start. In this case the game stops and $\exists$loise wins.

If $\exists$loise cannot find a suitable Y, the game stops and $\forall$belard wins. If $\exists$loise does find a suitable Y (one that is not covered by the halting clause in item (3) above) then Y is added to the list of played sets, and play continues. $\forall$belard must now choose a defect $\Diamond\psi$ from the last played Hintikka set with the following restriction: in round r he can only choose defects $\Diamond\psi$ such that the modal depth of $\Diamond\psi$ is less than or equal to the modal depth of φ minus r. $\exists$loise must respond as before. She wins if she can survive all his challenges (in other words, he loses if he reaches a situation where he cannot choose any more defects).

The φ-game stops after at most modal depth of φ many rounds. The information on the board is at any stage of the game polynomial in the length of φ. We claim that $\exists$loise has a winning strategy iff φ is satisfiable.

The right-to-left direction is clear: $\exists$loise has a winning strategy if φ is satisfiable, for she need simply play by reading the required Hintikka sets off the model. For the other direction, suppose $\exists$loise has a winning strategy for the φ-game. We create a model $\mathcal{M}$ for φ as follows. The domain M is built in steps by following her winning strategy. M_0 consists of her initial move $\{X_0, \dots, X_n\}$. Suppose M_j is defined. Then M_{j+1} consists of a copy of those Hintikka sets she plays when using her winning strategy for each of $\forall$belard's possible moves played in the Hintikka sets from M_j (except when she plays a Hintikka set from her initial move, then of course we do not make a copy). Let M be the disjoint union of all M_j for j smaller than the modal depth of φ. Set $R(m, m')$ iff for all $\Diamond\psi \in \mathsf{SF}(\varphi)$, $\Diamond\psi \notin m \Rightarrow \psi \notin m'$ holds, and set $V(p) = \{m \in M \mid p \in m\}$. The rules of the game guarantee that nominals are interpreted as singletons.

We claim that the following truth-lemma holds. For all $m \in M$ which she plays in round j (i.e., $m \in M_j$), for all ψ of modal depth less than or equal to the modal depth of φ minus j, $\mathcal{M}, m \models \psi$ if and only if $\psi \in m$. We only discuss the case of $\Diamond$, if $\Diamond\psi \in m$, then $\forall$belard challenged this defect, so $\exists$loise could respond with an m' containing ψ. Since for all $\Diamond\psi \in \mathsf{SF}(\varphi)$, $\Diamond\psi \notin m \Rightarrow \psi \notin m'$ holds, we have $R(m, m')$ and by induction hypothesis $\mathcal{M}, m \models$

$\Diamond\psi$. If $\Diamond\psi \notin m$ but $R(m, m')$ holds, then by our definition of R, $\psi \notin m'$, so again $\mathcal{M}, m \not\models \Diamond\psi$. Since $\exists$loise plays a Hintikka set containing φ in the first round, $\mathcal{M}$ satisfies φ. ❑

Since satisfiability of basic modal formulas on the class of all frames is already PSPACE-complete, we can conclude that, in this case, the addition of nominals does not increase the complexity of the satisfiability problem (up to a polynomial). This is not always the case:

PROPOSITION 34. *$\mathcal{H}$-satisfiability on the class of symmetric frames is* EXPTIME*-complete.*

Proof. For any modal formula φ, let $\varphi' = i \wedge \Diamond\neg i \wedge \Box\Box\Diamond i \wedge \Box(\neg i \to \varphi^{\neg i})$, where i is any nominal and $\varphi^{\neg i}$ is obtained from φ by relativising all modalities with $\neg i$ (that is, $\Diamond\varphi$ becomes $\Diamond(\neg i \wedge \varphi)$ and $\Box\varphi$ becomes $\Box(\neg i \to \varphi)$). It can be easily seen that if φ' holds at a world w in a symmetric model $\mathcal{M}$ then φ holds globally in the submodel of $\mathcal{M}$ generated by w, minus the world w itself. Conversely, a symmetric model on which φ holds globally can easily be turned into a model for φ'. It follows that, on symmetric frames, φ' is satisfiable iff φ is globally satisfiable.

The global satisfiability problem for modal formulas on the class of symmetric frames is known to be EXPTIME-complete [50]. Hence, the satisfiability problem for $\mathcal{H}$ on the class of symmetric frames is EXPTIME-hard. That the problem is inside EXPTIME will follow from Theorem 39. ❑

Note that the proof uses only a single nominal. The satisfiability problem for the modal logic of the class of symmetric frames, **KB**, is only PSPACE-complete [50]. Hence, assuming PSPACE $\neq$ EXPTIME, adding a single nominal already makes the satisfiability problem more complex. A similar blowup holds for tense logic: the satisfiability problem of the basic temporal logic is PSPACE, but the addition of a single nominal moves the complexity to EXPTIME. The complexity drops though when considering linear or branching time models (to NP-complete in the first case and to PSPACE-complete in the second) [7, 110].

Adding nominals can even result in logics that are undecidable and lack the finite model property, as was first observed in the context of description logics [100, 103]. The example below is taken from [20]. Consider the bi-modal language with modalities $\langle R_1\rangle$ and $\langle R_2\rangle$, and let KB23 be the frame class defined by the following modal Sahlqvist formulas:

$$\begin{array}{ll}
\bigwedge_{1\leq k\leq 3}\langle R_1\rangle p_k \to \bigvee_{1\leq k<l\leq 3}\langle R_1\rangle(p_k \wedge p_l) & \text{(at most 2 } R_1\text{-successors)}\\
\bigwedge_{1\leq k\leq 4}\langle R_1\rangle\langle R_1\rangle p_k \to \bigvee_{1\leq k<l\leq 4}\langle R_1\rangle\langle R_1\rangle(p_k \wedge p_l) & \text{(at most 3 two-step } R_1\text{-successors)}\\
p \to [R_2]\langle R_2\rangle p & (R_2 \text{ is symmetric}).
\end{array}$$

PROPOSITION 35. *The modal logic of* KB23 *has the finite model property and is decidable.*

Proof. First, consider the mono-modal logic axiomatized by the first two axioms. This logic is complete for a class of frames that is closed under taking subframes, and it has the bounded width property: no point has more than two successors. It follows that this logic has the finite model property and is decidable. Second, consider the mono-modal logic given by the last axiom. This logic, which is complete for the class of symmetric frames, has the finite model property [49] and its satisfiability problem is complete for PSPACE [50]. Since decidability and the finite model property are preserved under taking fusions [69], the result follows. ❑

PROPOSITION 36. *The $\mathcal{H}$-logic of* KB23 *is undecidable and lacks the finite model property.*

Proof. For any mono-modal formula φ with modality $\langle R_1\rangle$, let $\varphi^* = i\wedge\langle R_2\rangle\neg i\wedge[R_2][R_1]\langle R_2\rangle i\wedge [R_2](\neg i \to \varphi^{\neg i})$. Here again $\varphi^{\neg i}$ is obtained from φ by relativising all modalities with $\neg i$ as above. By the same argument as in the proof of Proposition 34, φ' is satisfiable on $\mathsf{KB23}$ iff φ is globally satisfiable on the class of (mono-modal) frames in which each point has at most two successors and at most three two-step successors. Global satisfiability of modal formulas on the latter frame class is undecidable [133]. It follows that the $\mathcal{H}$-logic of $\mathsf{KB23}$ is also undecidable, and hence, since it is recursively enumerable (as follows from the elementarity of $\mathsf{KB23}$), it lacks the finite model property. ❑

Next, let us consider the language $\mathcal{H}(@,\downarrow)$. As was observed in [6], $\mathcal{H}(@,\downarrow)$ is a *conservative reduction class* of first-order logic. Following [39], we call a fragment of first-order logic a conservative reduction class if there is a computable translation τ mapping first-order formulas to formulas in the fragment, such that for all formulas α, $\tau(\alpha)$ is satisfiable iff α is, and $\tau(\alpha)$ has a finite model iff α has. Every conservative reduction class has an undecidable (in fact Π^0_1-complete) satisfiability problem, as well as an undecidable (in fact Σ^0_1-complete) finite satisfiability problem [39].

THEOREM 37. *$\mathcal{H}(@,\downarrow)$ is a conservative reduction class.*

Proof. The class of first-order formulas with equality in a single binary relation is known to be a conservative reduction class [39]. Now, consider the following translation from this first-order language to $\mathcal{H}(@,\downarrow)$, where i is a fixed nominal:

$$\begin{array}{lcl}
(R(x,y))^* & = & @_x\langle R\rangle y \\
(x=y)^* & = & @_x y \\
(\neg\varphi)^* & = & \neg\varphi^* \\
(\varphi\wedge\psi)^* & = & \varphi^*\wedge\psi^* \\
(\exists x.\varphi)^* & = & @_i\langle R\rangle{\downarrow}x.(\varphi^*).
\end{array}$$

Clearly, $(\cdot)^*$ is a computable function. Moreover, a first-order sentence φ is satisfiable (in a finite model) iff φ^* is satisfiable (in a finite model). First, suppose $\mathcal{M}\models\varphi$. Let $\mathcal{M}'$ be the model obtained from $\mathcal{M}$ by adding a new state w, labelled with nominal i, extending the relation R such that $(w,v)\in R$ for all states v in the domain of M. Then, clearly, $\mathcal{M}'\models\varphi^*$. Moreover, $\mathcal{M}'$ is finite if M is. Conversely, suppose $\mathcal{M},w\models\varphi^*$. Let v be the state in $\mathcal{M}$ labelled by the nominal i, and let $\mathcal{M}'$ be the submodel of $\mathcal{M}$ whose domain consists of all successors of v. Then, clearly, $\mathcal{M}'\models\varphi$. Moreover, $\mathcal{M}'$ is finite if $\mathcal{M}$ is. ❑

Even though the satisfiability problem for $\mathcal{H}(@,\downarrow)$ is undecidable, in certain cases $\mathcal{H}(@,\downarrow)$ is still computationally more attractive than the full first-order language. For instance, the satisfiability problem for $\mathcal{H}(@,\downarrow)$ becomes decidable if we restrict the out-degree of the nodes in the model [139]. Figure 6 lists the results for mono-modal formulas. Here, for a given κ, we consider the class of frames were every node has strictly less than κ successors. In particular, if $\kappa=\omega$, then each state can only have finitely many successors, and if $\kappa=1$, the relation is the empty relation.

The undecidability of $\mathcal{H}(@,\downarrow)$ does not depend on the presence of nominals or propositional variables: even without these, the satisfiability problem is undecidable. Similarly, the undecidability does not depend on nested occurrences of $\downarrow$. One successful way to syntactically restrict the language in order to obtain decidability, is to restrict the interaction between $\downarrow$ and the modalities [105, 139]. In particular, it was proved in [139] that decidability is regained when formulas

Nr. of successors	$\mathcal{H}(@, \downarrow)$	*First-order correspondence language*
$\kappa = 1$	NP-complete	NEXPTIME-complete
$\kappa = 2$	NP-complete	Non-elementary decidable
$3 \leq \kappa < \omega$	NEXPTIME-complete	Π^0_1-complete (co-r.e., not decidable)
$\kappa = \omega$	Σ^0_1-complete (r.e., not decidable)	Σ^1_1-complete (highly undecidable)
$\kappa > \omega$	Π^0_1-complete (co-r.e., not decidable)	Π^0_1-complete (co-r.e., not decidable)

Figure 6. Complexity of the satisfiability problem on mono-modal models with bounded out-degree

of the form $\cdots \Box(\cdots {\downarrow}x.(\cdots \Box \cdots)\cdots)\cdots$ are excluded. In other words, the undecidability of $\mathcal{H}(@, \downarrow)$ is caused by formulas that, when put in negation normal form, contain a $\downarrow$-binder that is both in the scope of a box operator and that contains in its scope a box operator. This result was shown to be tight [139].

To round off this section, we will discuss two useful complexity results that can be used to prove upper bounds for the complexity of various hybrid logics: the *loosely $\forall$-bounded fragment with constants* and the *hybrid μ-calculus*.

Consider any first-order language not containing function symbols, but possibly containing constants. A formula of such a language is called *loosely $\forall$-guarded* if it is built up from possibly negated atomic formulas using conjunction, disjunction, existential quantification and loosely guarded universal quantification, i.e., universal quantification of the form $\forall\vec{x}(\varphi \to \psi)$, where $\vec{x}$ is a sequence of variables and φ is an atomic formula containing all free variables of ψ.

THEOREM 38 ([82, 138]). *The satisfiability problem for loosely $\forall$-guarded first-order formulas is* 2EXPTIME*-complete. It is* EXPTIME*-complete when there is a uniform bound on the number of variables occurring in the formula (but not necessarily on the number of constants).*

Many hybrid logics can be translated into the loosely $\forall$-guarded fragment using only a limited number of variables. For such logics, Theorem 38 provides an EXPTIME upper bound.

The hybrid μ-calculus [124] extends the modal μ-calculus (cf. Chapter 12 of this handbook) with nominals, converse operators and the universal modality. It expressively subsumes many propositional dynamic and temporal logics, such as (hybrid) PDL and CTL. Sattler and Vardi [124] showed by means of automata that the satisfiability problem for the hybrid μ-calculus is EXPTIME-complete. Beside the fact that this result singles out a very expressive hybrid language that is still decidable in EXPTIME, it is interesting because the proof is based on tree automata. For any formula φ, an automaton A_φ on infinite trees is given that accepts precisely the "tree models" of φ. Checking whether φ is satisfiable then reduces to solving the emptiness problem for A_φ. The catch, in the case of the hybrid μ-calculus, is that the standard tree model property fails for this language. The key idea in the proof is that a model of a hybrid μ-formula φ can be transformed into a forest by properly choosing points to witness diamond formulas. See [124] for details.

Below, we will give instead an alternative proof by providing a polynomial time satisfiability preserving translation of the full hybrid μ-calculus into its nominal-free fragment. But first, let us review the syntax and semantics of the language. The hybrid μ-calculus makes use of set variables, which we will write as $x, y, \ldots$, and which should not be confused with the state variables of hybrid languages such as $\mathcal{H}(@, \downarrow)$. The syntax is defined by the following inductive

definition[10]:

$$\varphi ::= p \mid i \mid x \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle R\rangle\varphi \mid \langle \bar{R}\rangle\varphi \mid \mathsf{E}\varphi \mid \mu x.\varphi,$$

where $p \in \mathsf{PROP}$, $i \in \mathsf{NOM}$, $R \in \mathsf{REL}$ and where x is a set variable occurring only positively in φ (i.e., under an even number of negation signs). Since the language contains set variables, the semantics is defined with the help of assignments. Here, an assignment will be a function g that assigns to each set variable a subset of the domain of the model. The semantics, then, is given by the following truth definition.

$\mathcal{M}, g, w \models p$	iff	$w \in V(p)$ for $p \in \mathsf{PROP} \cup \mathsf{NOM}$
$\mathcal{M}, g, w \models x$	iff	$w \in g(x)$
$\mathcal{M}, g, w \models \neg\varphi$	iff	$\mathcal{M}, g, w \not\models \varphi$
$\mathcal{M}, g, w \models \varphi \wedge \psi$	iff	$\mathcal{M}, g, w \models \varphi$ and $\mathcal{M}, g, w \models \psi$
$\mathcal{M}, g, w \models \langle R\rangle\varphi$	iff	there is a $v \in W$ such that $R(w, v)$ and $\mathcal{M}, g, v \models \varphi$
$\mathcal{M}, g, w \models \langle \bar{R}\rangle\varphi$	iff	there is a $v \in W$ such that $R(v, w)$ and $\mathcal{M}, g, v \models \varphi$
$\mathcal{M}, g, w \models \mathsf{E}\varphi$	iff	there is a $v \in W$ such that $\mathcal{M}, g, v \models \varphi$
$\mathcal{M}, g, w \models \mu x.\varphi$	iff	for all $W' \subseteq W$, if $\{v \in W \mid \mathcal{M}, g^x_{W'}, v \models \varphi\} \subseteq W'$ then $w \in W'$.

THEOREM 39 ([124]). *The satisfiability problem for the hybrid μ-calculus is* EXPTIME-*complete.*

Proof. We define a polynomial time satisfiability preserving translation from the full hybrid μ-calculus to its nominal-free fragment, i.e., the modal μ-calculus with converse operators and the existential modality. Since the latter language is EXPTIME complete [143, 48], the result follows.

Consider any formula φ of the hybrid μ-calculus containing nominals $i_1, \ldots, i_n$. For each nominal i_k, introduce a new distinct propositional variable q_k. In the translation we will define, each nominal will be uniformly replaced by the corresponding propositional variable. Clearly, we cannot force these propositional variables to denote singleton sets. We *can*, however, ensure that the formula in question does not distinguish between states named by the same nominal. To this end, we will use $\langle\equiv\rangle\psi$ as a shorthand for the formula $\mu x.(\psi \vee \bigvee_{k\leq n}(q_k \wedge \mathsf{E}(q_k \wedge x)))$, which says that ψ holds either at the current state, or at a state satisfying the same nominal as the current state, or in general at any state reachable from the current world in finitely many steps along the "satisfies the same nominal" relation. Now, define φ^* inductively, as follows:

$$\begin{array}{rcl}
(i_k)^* &=& \langle\equiv\rangle q_k \\
p^* &=& \langle\equiv\rangle p \\
x^* &=& \langle\equiv\rangle x \\
(\neg\psi)^* &=& \neg\psi^* \\
(\psi \wedge \chi)^* &=& \psi^* \wedge \chi^* \\
(\langle R\rangle\psi)^* &=& \langle\equiv\rangle\langle R\rangle\langle\equiv\rangle\psi \\
(\langle \bar{R}\rangle\psi)^* &=& \langle\equiv\rangle\langle \bar{R}\rangle\langle\equiv\rangle\psi \\
(\mathsf{E}\psi)^* &=& \mathsf{E}\psi^* \\
(\mu x.\psi)^* &=& \mu x.\psi^*.
\end{array}$$

Finally, let $\varphi^+ = \varphi^* \wedge \bigwedge_{k\leq n} \mathsf{E}p_k$. Note that φ^+ does not contain any nominals, and is only polynomially longer than φ. We will now show that φ and φ^+ are equi-satisfiable. One direction is trivial: if $\mathcal{M}, w \models \varphi$, then, assigning to each q_k the same (singleton) denotation

[10] Our notation is slightly different from the one used in [124].

as i_k, we obtain that $\mathcal{M}, w \models \varphi^+$ (note that, in this case, $\equiv$ is the identity relation). Conversely, suppose $\mathcal{M}, w \models \varphi^+$, with $\mathcal{M} = \langle W, (R^{\mathcal{M}})_{R \in \mathsf{REL}}, V \rangle$. Let $\equiv$ be the smallest equivalence relation on W such that $v \equiv u$ whenever v and u both satisfy q_k for some $k \leq n$. Let $\widehat{\mathcal{M}} = \langle W/_{\equiv}, (R^{\widehat{\mathcal{M}}})_{R \in \mathsf{REL}}, V' \rangle$, where $W/_{\equiv}$ is the set of $\equiv$-equivalence classes of W,

$$\begin{array}{lll} R^{\widehat{\mathcal{M}}}([v],[u]) & \text{iff} & \text{there are } v' \in [v] \text{ and } u' \in [u] \text{ such that } R^{\mathcal{M}}(v',u'), \\ [v] \in V'(p) & \text{iff} & \text{there is a } v' \in [v] \text{ such that } v' \in V(p), \text{ and} \\ [v] \in V'(i_k) & \text{iff} & \text{there is a } v' \in [v] \text{ such that } v' \in V(q_k). \end{array}$$

By construction, $V'(i_k)$ is a singleton set for each $k \leq n$. Moreover, it follows directly from the definition of $(\cdot)^*$ and $\widehat{\mathcal{M}}$ that, for all formulas ψ of the hybrid μ-calculus, and for all worlds $v \in W$,

$$\widehat{\mathcal{M}}, [v] \models \psi \quad \text{iff} \quad \mathcal{M}, v \models \psi^*.$$

In particular, $\widehat{\mathcal{M}}, [w] \models \varphi$. ❑

5 PROOF THEORY

In this section we discuss proof methods for hybrid logics, and show examples of how to use them. We will first present two "classical" proof systems (a sequent calculus and a natural deduction calculus), and then two others (a tableau calculus and a resolution calculus), which are usually considered more suitable for implementations. We will focus on the languages $\mathcal{H}(@)$ and $\mathcal{H}(@, \downarrow)$.

5.1 Sequent Calculus

The first modern results on proof theory for hybrid logics can be found in the work of Seligman in the area of Situation Theory [128, 129]. This work deals with strong ($\forall$-based) systems, but many of the key ideas underlying hybrid deduction (in particular, the deductive significance of @) were first explored in these papers.

The calculus $\mathbf{S}_{\mathcal{H}(@,\downarrow)}$ in Figure 7 is from [130] where a sound and complete sequent calculus for hybrid logics is developed from a sequent calculus for first-order logics by a series of transformations. In the figure, $\Gamma \vdash \Delta$ is a sequent where Γ and Δ are sets of hybrid formulas and φ, Γ is taken to be $\{\varphi\} \cup \Gamma$. The techniques used are quite general and can be applied to a wide range of hybrid and modal logics. Notice that the calculus is cut free. It can be proved that the cut rule is admissible.

An interesting feature of $\mathbf{S}_{\mathcal{H}(@,\downarrow)}$ is that the calculus is not restricted to @-formulas, as the other calculus we are going to discuss in the following sections. Intuitively, an @-prefixed sequent calculus can be generalized to deal with all formulas by using nominals as follows. A single nominal a on the left side of a sequent is enough to anchor all non @-prefixed formulas to the same element and so removes the need for them to share an @ prefix. The price to pay for this is that the calculus does not have a subformula property, as a proof may contain any number of @-prefixes which are not present in the end sequent (introduced using the $^\wedge$@ rules). But it is easy to prove that only "one layer" of prefixes is needed in any proof, and define a version of the subformula property that takes this into account.

The presence of nominals and the @ operator in the calculus above is crucial. When the underlying modal logic is temporal logic, more flexibility is possible: Demri [57] presents a sequent system for nominal tense logic, which does not contains @.

$$
\begin{array}{cccc}
 & (I) \quad \dfrac{}{\varphi, \Gamma \vdash \Delta, \varphi} & & \\[2ex]
 & (N_L) \quad \dfrac{a, b, \Gamma[a] \vdash \Delta[a]}{a, b, \Gamma[b] \vdash \Delta[b]} & & \\[2ex]
(\neg_L) & \dfrac{\Gamma \vdash \Delta, \varphi}{\neg\varphi, \Gamma \vdash \Delta} & (\neg_R) & \dfrac{\varphi, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg\varphi} \\[2ex]
(\vee_L) & \dfrac{\varphi, \Gamma \vdash \Delta \quad \psi, \Gamma \vdash \Delta}{(\varphi \vee \psi), \Gamma \vdash \Delta} & (\vee_R) & \dfrac{\Gamma \vdash \Delta, \varphi, \psi}{\Gamma \vdash \Delta, (\varphi \vee \psi)} \\[2ex]
(\langle r\rangle_L)^1 & \dfrac{\langle r\rangle a, @_a\varphi, \Gamma \vdash \Gamma}{\langle r\rangle\varphi, \Gamma \vdash \Delta} & (\langle r\rangle_R) & \dfrac{\Gamma \vdash \Delta, @_a\varphi \quad \Gamma \vdash \Delta, \langle r\rangle a}{\Gamma \vdash \Delta, \langle r\rangle\varphi} \\[2ex]
(\downarrow_L) & \dfrac{a, \varphi[x/a], \Gamma \vdash \Delta}{a, \downarrow x.\varphi, \Gamma \vdash \Delta} & (\downarrow_R) & \dfrac{a, \Gamma \vdash \Delta, \varphi[x/a]}{a, \Gamma \vdash \Delta, \downarrow x.\varphi} \\[2ex]
({}^{\vee}@_L) & \dfrac{a, \varphi, \Gamma \vdash \Delta}{a, @_a\varphi, \Gamma \vdash \Delta} & ({}^{\vee}@_R) & \dfrac{a, \Gamma \vdash \Delta, \varphi}{a, \Gamma \vdash \Delta, @_a\varphi} \\[2ex]
({}^{\wedge}@_L) & \dfrac{a, @_a\varphi, \Gamma \vdash \Delta}{a, \varphi, \Gamma \vdash \Delta} & ({}^{\wedge}@_R) & \dfrac{a, \Gamma \vdash \Delta, @_a\varphi}{a, \Gamma \vdash \Delta, \varphi}
\end{array}
$$

$$
(\text{name})^2 \quad \dfrac{a, \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \qquad (\text{term})^3 \quad \dfrac{a, \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \qquad (\text{term}^-)^3 \quad \dfrac{\Gamma \vdash \Delta}{a, \Gamma \vdash \Delta}
$$

Restrictions:

[1] if a does not occur in φ, Γ, Δ.

[2] if a does not occur in Γ, Δ.

[3] if all formulas in Γ, Δ are @-prefixed.

Figure 7. Rules for the Sequent Calculus $\mathbf{S}_{\mathcal{H}(@,\downarrow)}$

EXAMPLE 40. We prove the sequent $\downarrow x.\langle R\rangle(x \wedge p) \vdash p$ in $\mathbf{S}_{\mathcal{H}(@,\downarrow)}$:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{b, @_a\langle R\rangle b, a, p \vdash p}\,(I)}{b, @_a\langle R\rangle b, a, p \vdash @_a p}\,({}^{\vee}@_R)}{b, @_a\langle R\rangle b, a \wedge p \vdash @_a p}\,(\neg_L), (\neg_R), \text{ and } (\vee_R)}{b, @_a\langle R\rangle b, @_b(a \wedge p) \vdash @_a p}\,({}^{\vee}@_L)}{@_a\langle R\rangle b, @_b(a \wedge p) \vdash @_a p}\,(\text{term})}{a, @_a\langle R\rangle b, @_b(a \wedge p) \vdash @_a p}\,(\text{term}^-)}{a, \langle R\rangle b, @_b(a \wedge p) \vdash p}\,({}^{\wedge}@_L) \text{ and } ({}^{\wedge}@_R)}{a, \langle R\rangle(a \wedge p) \vdash p}\,(\langle R\rangle_L)}{a, \downarrow x.\langle R\rangle(x \wedge p) \vdash p}\,(\downarrow_L)}{\downarrow x.\langle R\rangle(x \wedge p) \vdash p}\,(\text{name})
$$

5.2 Natural Deduction Calculus

Seligman proposed also a natural deduction system (again, not restricted to @-formulas) in [129]. However, the paper only proves soundness and completeness and does not discuss whether the calculus is normalizing. Braüner introduced in [42] an @-prefixed natural deduction calculus for $\mathcal{H}(@, \downarrow, \forall)$ and its sublanguages and established normalization. Figure 8 shows the rules corresponding to the $\mathcal{H}(@, \downarrow)$ fragment.

$$(\wedge_I)\ \frac{@_a\varphi \quad @_a\psi}{@_a(\varphi \wedge \psi)} \qquad (\wedge_{E1})\ \frac{@_a(\varphi \wedge \psi)}{@_a\varphi} \qquad (\wedge_{E2})\ \frac{@_a(\varphi \wedge \psi)}{@_a\psi}$$

$$(\rightarrow_I)\ \frac{\begin{array}{c}[@_a\varphi]\\ \vdots\\ @_a\psi\end{array}}{@_a(\varphi \rightarrow \psi)} \qquad (\rightarrow_E)\ \frac{@_a(\varphi \rightarrow \psi) \quad @_a\varphi}{@_a\psi}$$

$$(\bot_1)^1\ \frac{\begin{array}{c}[@_a\neg\varphi]\\ \vdots\\ @_a\bot\end{array}}{@_a\varphi} \qquad (\bot_2)\ \frac{@_a\bot}{@_c\bot}$$

$$(@_I)\ \frac{@_a\varphi}{@_c@_a\varphi} \qquad (@_E)\ \frac{@_c@_a\varphi}{@_a\varphi}$$

$$([r]_I)^2\ \frac{\begin{array}{c}[@_a\langle r\rangle c]\\ \vdots\\ @_c\varphi\end{array}}{@_a[r]\varphi} \qquad ([r]_E)\ \frac{@_a[r]\varphi \quad @_a\langle r\rangle c}{@_c\varphi}$$

$$(\downarrow_I)^3\ \frac{\begin{array}{c}[@_a c]\\ \vdots\\ @_c\varphi[x/c]\end{array}}{@_a{\downarrow}x.\varphi} \qquad (\downarrow_E)\ \frac{@_a{\downarrow}x.\varphi \quad @_a c}{@_c\varphi[x/c]}$$

$$(\text{Ref})\ \frac{}{@_a a} \qquad (\text{Nom}_1)^4\ \frac{@_a c \quad @_a\varphi}{@_c\varphi} \qquad (\text{Nom}_2)\ \frac{@_a c \quad @_a\langle r\rangle b}{@_c\langle r\rangle b}$$

Restrictions:

1. φ is a propositional variable.
2. c is not free in $@_a[r]\varphi$ or in any undischarged assumptions other than the specified occurrences of $@_a\langle r\rangle c$.
3. c is not free in $@_a{\downarrow}x.\varphi$ or in any undischarged assumptions other than the specified occurrences of $@_a c$.
4. φ is a propositional variable or a nominal.

Figure 8. Rules for the natural deduction calculus $\mathbf{ND}_{\mathcal{H}(@,\downarrow)}$

The system $\mathbf{ND}_{\mathcal{H}(@,\downarrow)}$ can be extended in a complete way with additional inference rules

corresponding to first-order conditions on the accessibility relations expressed by *geometric theories*[11]. And as we said, the system $\mathbf{ND}_{\mathcal{H}(@,\downarrow)}$ enjoys normalization (even when extended with rules for geometric theories), and a suitable version of the subformula property that takes into account the use of @-formulas. See [42], for further details. In [43] Braüner compares his system with Seligman's (actually, a slight variation of Seligman's to ensure closure under substitutions), providing translations of proofs in both directions. These translations allows us to transfer reduction rules between Braüner's and Seligman's calculus, but they are not sufficient to ensure normalization of the latter. Hence, the normalization problem for Seligman's calculus is still open.

EXAMPLE 41. We prove that $\downarrow x.\langle R\rangle(x \wedge p) \to p$ is a tautology in $\mathbf{ND}_{\mathcal{H}(@,\downarrow)}$:

$$
\begin{prooftree}
\AxiomC{$[@_y(\downarrow x.\langle R\rangle(x\wedge p))]^1$}
\AxiomC{}
\RightLabel{(Ref)}
\UnaryInfC{$@_y y$}
\RightLabel{$(\downarrow_E)$}
\BinaryInfC{$@_y\langle R\rangle(y\wedge p)$}
\AxiomC{$[@_y\neg p]^2$}
\AxiomC{$[@_z(y\wedge p)]^3$}
\RightLabel{$(\wedge_{E1})$}
\UnaryInfC{$@_z y$}
\AxiomC{$[@_z(y\wedge p)]^3$}
\RightLabel{$(\wedge_{E2})$}
\UnaryInfC{$@_z p$}
\RightLabel{(Nom_1)}
\BinaryInfC{$@_y p$}
\RightLabel{$(\to_E)$}
\BinaryInfC{$@_y\bot$}
\RightLabel{$(\bot_2)$}
\UnaryInfC{$@_z\bot$}
\RightLabel{$(\to_I)^3$}
\UnaryInfC{$@_z\neg(y\wedge p)$}
\RightLabel{$([R]_E)$}
\UnaryInfC{$@_y[R]\neg(y\wedge p)$}
\RightLabel{$(\to_E)$}
\BinaryInfC{$@_y\bot$}
\RightLabel{$(\bot_1)^2$}
\UnaryInfC{$@_y p$}
\RightLabel{$(\to_I)^1$}
\UnaryInfC{$@_y((\downarrow x.\langle R\rangle(x\wedge p))\to p)$}
\end{prooftree}
$$

5.3 *Tableau Calculus*

In Figure 9 the rules for a tableau calculus for $\mathcal{H}(@,\downarrow)$ are given. This calculus was introduced in [29], where tableau calculi for a family of *quantified* hybrid logics are presented (these are extensions of the propositional calculus defined in [25]). As in the case of natural deduction, the calculus is @-based: to prove the unsatisfiability of φ, apply the rules in Figure 9 to $@_i\varphi$ for i a nominal not in φ. If a closed tableau is found (i.e., a tableau in which each branch contains a pair of formulas $@_j\psi$ and $@_j\neg\psi$), then the original formula is unsatisfiable.

Completeness of the tableau calculus is proved for frame classes that can be axiomatized by pure, nominal free hybrid sentences[12]. Moreover, the tableau calculus can be used for effectively computing interpolants for a pair of formulas φ, ψ such that $\varphi \to \psi$ is a validity. The following result is proved in [30] using Fitting's argument for proving the same property for first-order logic [63].

THEOREM 42. *Given a closed hybrid tableau for $\varphi \to \psi$ using the rules of $\mathbf{T}_{\mathcal{H}(@,\downarrow)}$, the interpolant can be computed effectively.*

[11] A first-order formula is geometric if it is built out of atomic formulas of the form $R(x,y)$ and $x = y$ using only the connectives $\bot$, $\wedge$, $\vee$ and $\exists$. A geometric theory is a finite set of closed first-order formulas each having the form $\forall\bar{x}(\varphi \to \psi)$ where the formulas φ and ψ are geometric.

[12] The completeness proof is interesting: a valid hybrid sentence is translated into a valid first-order sentence in the correspondence language for which first-order closed tableau should exist; the tableau proof is then translated back into a hybrid tableau proof.

Constant Rules:	$(\neg\bot)\ \dfrac{\neg @_s \bot}{\top}$	$(\neg\top)\ \dfrac{\neg @_s \top}{\bot}$	
Negation Rules:	$(@)\ \dfrac{@_s \neg\varphi}{\neg @_s \varphi}$	$(\neg @)\ \dfrac{\neg @_s \neg\varphi}{@_s \varphi}$	
Conjunctive rules:	$(\wedge)\ \dfrac{@_s(\varphi \wedge \vee)}{\begin{array}{c}@_s\varphi\\ @_s\psi\end{array}}$	$(\neg\vee)\ \dfrac{\neg @_s(\varphi \vee \psi)}{\begin{array}{c}\neg @_s\varphi\\ \neg @_s\psi\end{array}}$	$(\neg \rightarrow)\ \dfrac{\neg @_s(\varphi \rightarrow \psi)}{\begin{array}{c}@_s\varphi\\ \neg @_s\psi\end{array}}$
Disjunctive rules:	$(\vee)\ \dfrac{@_s(\varphi \vee \psi)}{@_s\varphi \mid @_s\psi}$	$(\neg\wedge)\ \dfrac{\neg @_s(\varphi \wedge \psi)}{\neg @_s\varphi \mid \neg @_s\psi}$	$(\rightarrow)\ \dfrac{@_s(\varphi \rightarrow \psi)}{\neg @_s\varphi \mid @_s\psi}$
Diamond Rules:	$(\langle r\rangle)\ \dfrac{@_s\langle r\rangle\varphi}{\begin{array}{c}@_s\langle r\rangle t\\ @_t\varphi\end{array}}$ for t new in branch	$(\neg[r])\ \dfrac{\neg @_s[r]\varphi}{\begin{array}{c}@_s\langle r\rangle t\\ \neg @_t\varphi\end{array}}$ for t new in branch	
Box Rules:	$([r])\ \dfrac{@_s[r]\varphi \quad @_s\langle r\rangle t}{@_t\varphi}$	$(\neg\langle r\rangle)\ \dfrac{\neg @_s\langle r\rangle\varphi \quad @_s\langle r\rangle t}{\neg @_t\varphi}$	
@ rules:	$(@)\ \dfrac{@_s @_t\varphi}{@_t\varphi}$	$(\neg @)\ \dfrac{\neg @_s @_t\varphi}{\neg @_t\varphi}$	
	(Ref) $\dfrac{[s \text{ on the branch}]}{@_s s}$	(Nom) $\dfrac{@_s t \quad @_s\varphi}{@_t\varphi}$	(Bridge) $\dfrac{@_s t \quad @_u\langle r\rangle s}{@_u\langle r\rangle t}$
Downarrow Rules:	$(\downarrow)\ \dfrac{@_s{\downarrow}x.\varphi}{@_s\varphi[s/x]}$	$(\neg\downarrow)\ \dfrac{\neg @_s{\downarrow}x.\varphi}{\neg @_s\varphi[s/x]}$	

Figure 9. Rules for the tableau calculus $\mathbf{T}_{\mathcal{H}(@,\downarrow)}$

In a slightly different direction, Tzakova [142] presents a general approach to hybrid tableaux using Fitting-style prefix calculi. Such tableau use nominals both as part of the object language and as meta-logical labels.

Tableau methods have played a crucial role in modern automated reasoning for modal logics, and the best state-of-the-art provers for modal-like logics (such as the description logics provers RACER [84, 83] or FACT [90, 89]) are based on tableaux (see Chapter 13 of this handbook for further details). A variation of the hybrid tableau calculus of Figure 9 has been equipped with heuristics to ensure termination in [38]. The ideas used are related to the techniques used for terminating tableaux for the description logic $\mathcal{SHOIQ}$ [92].

EXAMPLE 43. We prove that ${\downarrow}x.(\langle R\rangle(x \wedge p) \rightarrow p)$ is a tautology in $\mathbf{T}_{\mathcal{H}(@,\downarrow)}$:

1.	$\neg @_i(\downarrow x.(\langle R\rangle(x \wedge p) \to p))$	Negation of the input formula
2.	$\neg @_i(\langle R\rangle(i \wedge p) \to p)$	$(\neg\downarrow)$ in 1
3.	$@_i(\langle R\rangle(i \wedge p)$	$(\neg \to)$ in 2
4.	$\neg @_i p$	$(\neg \to)$ in 2
5.	$@_i\langle R\rangle j$	$(\langle R\rangle)$ in 3
6.	$@_j(i \wedge p)$	$(\langle R\rangle)$ in 3
7.	$@_j i$	$(\wedge)$ in 6
8.	$@_j p$	$(\wedge)$ in 6
9.	$@_i p$	(Nom) in 7 and 8
	$\times$	Clash between 4 and 9

5.4 Resolution Calculus

As we just mentioned, the most successful automated theorem proving implementations for modal logics are based on the tableau method. Much of their outstanding performance is due to the heavy use of several heuristics and optimizations [93]; however, a number of these techniques do not work when the underlying logic allows some form of equality as in the case of hybrid logics. When nominals and satisfaction operators are added, the performance of tableau-based theorem provers is affected. This motivated research on possible alternatives, such as the resolution calculus. The best automated theorem provers for first-order logic are based on resolution, and we have already seen many similarities between hybrid and first-order logics.

Resolution calculi for $\mathcal{H}(@, \downarrow)$ and its sublanguages were introduced in [10, 11]. In a recent paper [14], the calculus for $\mathcal{H}(@)$ was refined to include ordering and selection functions (see [17] for the definitions of these standard notions). The rules are shown in Figure 10. In the figure, $S(C)$ is a selection function and $\succ$ is an admissible order; furthermore, the main premise of each rules is on the right. The calculus works on formulas in negation normal form (i.e., negation can only appear on atomic formulas), and hence an explicit rule for negation is not required. To extend the calculus to $\mathcal{H}(@, \downarrow)$, simply add the rule

$$(\downarrow) \qquad \frac{Cl \cup \{@_t{\downarrow} x.\varphi\}}{Cl \cup \{@_t\varphi[x/t]\}}.$$

Given a formula φ (in negation normal form), let *ClSet*$(\varphi) = \{\{@_i\varphi\}\}$, where i is a nominal not occurring in φ. Define *ClSet*$^*(\varphi)$ — the saturated set of clauses for φ — as the smallest set that includes *ClSet*(φ) and is saturated under the rules of Figure 10 (where saturation means that whenever there are sets matching the antecedent of any rule in *ClSet*$^*(\varphi)$ then also the sets in the consequent should be in *ClSet*$^*(\varphi)$). Then φ is unsatisfiable if and only if $\{\} \in$ *ClSet*$^*(\varphi)$.

The calculus $\mathbf{R}_{\mathcal{H}(@)}$ is implemented in the automated theorem prover HyLoRes [15], which uses an ordering that ensures termination while preserving soundness and completeness.

EXAMPLE 44. We prove that $\downarrow x.\langle R\rangle(x \wedge p) \to p$ is a tautology in $\mathbf{R}_{\mathcal{H}(@,\downarrow)}$. Consider the clause set corresponding to the negation of the formula:

$$(\wedge)\ \ \frac{Cl \cup \{@_t(\varphi_1 \wedge \varphi_2)\}}{\begin{array}{c}Cl \cup \{@_t\varphi_1\}\\ Cl \cup \{@_t\varphi_2\}\end{array}} \qquad (\vee)\ \ \frac{Cl \cup \{@_t(\varphi_1 \vee \varphi_2)\}}{Cl \cup \{@_t\varphi_1, @_t\varphi_2\}}$$

$$(\mathrm{RES})\ \ \frac{Cl_1 \cup \{@_t\varphi\} \quad Cl_2 \cup \{@_t\neg\varphi\}}{Cl_1 \cup Cl_2}$$

$$([r])\ \ \frac{Cl_1 \cup \{@_t\langle r\rangle s\} \quad Cl_2 \cup \{@_t[r]\varphi\}}{Cl_1 \cup Cl_2 \cup \{@_s\varphi\}} \qquad (\langle r\rangle)\ \ \frac{Cl \cup \{@_t\langle r\rangle\varphi\}}{\begin{array}{c}Cl \cup \{@_t\langle r\rangle n\}\\ Cl \cup \{@_n\varphi\}\end{array}}\ \ \begin{array}{l}\text{for } n \text{ a new nominal}\\ \text{and } \varphi \notin \mathsf{NOM}\end{array}$$

$$(@)\ \ \frac{Cl \cup \{@_t@_s\varphi\}}{Cl \cup \{@_s\varphi\}} \qquad (\mathrm{REF})\ \ \frac{Cl \cup \{@_t\neg t\}}{Cl}$$

$$(\mathrm{SYM})\ \ \frac{Cl \cup \{@_s t\}}{Cl \cup \{@_t s\}}\ \text{if } t \succ s \qquad (\mathrm{PARAM})\ \ \frac{Cl_1 \cup \{@_s t\} \quad Cl_2 \cup \{\varphi(s)\}}{Cl_1 \cup Cl_2 \cup \{\varphi(s/t)\}}\ \ \begin{array}{l}\text{if } s \succ t \text{ and}\\ \varphi(s) \succ @_s t\end{array}$$

Restrictions: Let φ and ψ be the displayed formulas in each of the above rules:

- Let $C = C' \cup \{\varphi\}$ be the main premise, then either $S(C) = \{\varphi\}$ or, otherwise, $S(C) = \emptyset$ and $\{\varphi\} \succ C'$.
- Let $D = D' \cup \{\psi\}$ be the auxiliary premise, then $\{\psi\} \succ D'$ and $S(D) = \emptyset$.

Figure 10. Rules for the resolution calculus $\mathbf{R}_{\mathcal{H}(@)}$ with Order and Selection Functions

1. $\{@_i((\downarrow x.\langle R\rangle(x \wedge p))\underline{\triangle}\neg p)\}$ by $(\wedge)$
2. $\{@_i\underline{\downarrow x}.\langle R\rangle(x \wedge p)\}, \{@_i\neg p\}$ by $(\downarrow)$
3. $\{@_i\underline{\langle R\rangle}(i \wedge p)\}, \{@_i\neg p\}$ by $(\langle r\rangle)$
4. $\{@_i\langle R\rangle j\}, \{@_j(i\underline{\triangle}p)\}, \{@_i\neg p\}$ by $(\wedge)$
5. $\{@_j\underline{i}\}, \{@_{\underline{j}}p\}, \{@_i\neg p\}$ by (PARAM)
6. $\{@_i\underline{p}\}, \{@_i\underline{\neg p}\}$ by (RES)
7. $\{\}$.

6 RELATION WITH OTHER FIELDS

In various areas, hybrid logics has been proposed as a convenient extension of modal logics, either because they give rise to smoother proof systems, or because of their greater expressive power. In this section we briefly discuss a number of cases, and provide pointers to the literature.

Temporal Logic. As indicated in the work of Prior and Bull, hybrid languages allow us to make explicit references to specific times (days, dates, years, etc.), and also to cope with temporal indexicals (such as yesterday, today, tomorrow and now). In addition, many temporally relevant frame properties (such as irreflexivity, asymmetry and trichotomy) that cannot be defined by means of modal formulas can be defined with nominals [28]. When nominals and satisfaction operators are added to an interval-based logic, the result is a $\mathsf{Holds}(t, \varphi)$-driven interval logic similar to those introduced in AI by James Allen [2] (where the satisfaction operators play the

role of Holds). By making explicit temporal references possible (combining nominals, satisfaction operators and temporal modalities, one can directly express temporal relations between instants or intervals), hybrid logics remove a serious obstacle to a modal analysis of temporal representation and reasoning.

Nominal tense logics have been studied in detail in [21]. The complexity of the satisfiability problem for a number of hybrid temporal logics is investigated in [7, 65]. The minimal hybrid tense logic $\mathcal{H}(\langle R^{-1}\rangle)$ is EXPTIME over the class of all frames and the class of transitive frames, but the complexity drops to NP-complete over the usual frames for linear time (strict total orders), and to PSPACE-complete over the usual frames for branching time (transitive trees). In [7, 110, 65], results are also given for hybrid languages with the Since and Until operators. Hybrid interval logics were recently studied in [95].

Indexicality and Direct Reference. Hybrid languages are also a powerful resource for studying indexicality in natural language, as an alternative to the more classical use of multi-dimensional modal logic. In the multi-dimensional modal approach, formulas are evaluated at sequences of points, where one point of the sequence is thought of as the point of evaluation, while the others are used as memory locations to store references [96, 146, 66, 52, 53]. Hybrid languages move multi-dimensional logic's sequence of evaluation points from the meta-language to the object language, with hybrid variables acting as names for indices (see [24]), and allowing in this way a natural treatment of such indexicals as 'today.' Moreover, when equipped with the @ operator, hybrid languages offer the 'de-scoping' behavior typical of such multi-dimensional operators as here and there. There are also links between hybrid logic and mathematical aspects of multi-dimensional modal logic, particularly the multi-dimensional modal perspective on cylindric algebra (cf. [106]), as $\downarrow$ and @ can be considered as explicit substitution devices.

Feature Logic. Most unification-based approaches to natural language grammar, such as PATR-II, use attribute value matrices (AVMs) to represent feature structures, where re-entrance in the feature structures is represented by "tags" in the AVMs [123]. There is a tight connection between AVMs and deterministic multi-modal logic, except that there is no clear way to express re-entrance in modal logic. As it turns out, the tags that are used to enforce re-entrance in AVMs correspond in a very natural way to nominals in hybrid logic. Thus, adding nominals is enough to make re-entrance expressible.

Previous approaches to encoding re-entrance in modal logic used more complicated techniques. In particular, Kasper-Rounds logic is essentially a fragment of deterministic propositional dynamic logic with program intersection, where the intersection is used to encoding re-entrance. See [33, 23, 120] for further details.

Dynamic Logic. As we discussed in Section 2.2, hybrid languages were rediscovered, many years after the work of Prior and Bull, by a group of logicians at the Sofia University in Bulgaria. Gargov, Passy and Tinchev were interested in neat axiomatizations of operators in PDL, and they realized that certain operators (e.g., union of programs) are easily captured, whereas others (e.g., program intersection or complement) require extra expressive power. In [113] it is shown that adding nominals is enough to enable natural and succinct characterization of these operators. Adding other kinds of "constants" to the language permits the representation of notions like determinism and looping [73]. In addition, the work of the Sofia school shows how nominals can be used to simplify the construction of models during completeness proofs [114]. See [115] for an excellent overview on combinatory dynamic logics.

For a modern discussion of PDL with nominals (in the framework of description logics) and some new complexity results see [56, 55].

Description Logics. Descriptions logics (DLs) are a family of formalisms that allow the representation of, and reasoning about, conceptual knowledge, in a structured and semantically well-understood manner [16]. They evolved from the original knowledge maintenance system KL-ONE of Brachman and Schmolze [41]. Description logics are discussed in detail in Chapter 13 of this handbook.

In [125] Schild identifies a close connections between description logics and modal logics, and uses it to transfer complexity and axiomatization results between the two areas. This connection is established at the level of concepts: *concepts* in description logic are shown to correspond to *formulas* in modal logic. Description logics, however, usually have two levels of representation. The first level is that of *concepts*, which, like modal formulas, denote subsets of the domain. The second level is that of *terminology boxes* (TBoxes) and *assertion boxes* (ABoxes). Using these, one can specify global conditions on models, such as the 'concept inclusion' $C \sqsubseteq D$, which requires that every individual satisfying the concept C should also satisfy the concept D, and the 'assertion' $a{:}C$, which requires that the individual a satisfies the concept C. The basic modal language is not rich enough to express such constructions. By lifting the correspondence to Converse PDL, Schild managed to account for inference with TBoxes. De Giacomo and Lenzerini [56, 55] further extended these results by encoding also ABoxes in Converse PDL.

While the embedding of DLs into Converse PDL have proved useful, it has two important disadvantages. Complexity-wise, the satisfiability problem of Converse PDL is already EXPTIME-complete and, hence, optimal complexity results cannot always be obtained with this technique. Moreover, the model theory of Converse PDL is complicated, due to the presence of the Kleene star (which requires a weak form of induction). Using the extended expressive power of hybrid languages, assertions can be encoded using satisfaction operators, and concept inclusions can be expressed using the universal modality A. See [12, 4, 36] for detailed discussions on the connections between hybrid and description logics.

Nominals have in fact been independently introduced in DLs. Very early systems like CLASSIC [40] and LOOM [104] already included a form of nominals in the late 80s. Such systems allowed a concept constructor called $\mathcal{O}$ (for "one-of") which permitted enumeration of individuals in the domain of a model. One-of expressions are in fact the same as disjunctions of nominals. The interest in the $\mathcal{O}$ operator dropped during the following years because of complexity issues (as we have seen in Section 4, the presence of nominals can lead to an increase in complexity, and even to undecidability, in the presence of other operators). However, the topic has recently regained interest, as direct reference to individuals seems to be a must for languages for the semantic web, one of the most important modern applications of DLs [94, 91]. The $\mathcal{O}$ operator is now part of the W^{3}C-recommended web ontology language OWL [107].

Information Systems. Nominals have turned up in yet another setting, namely the Polish tradition of modal logics for information systems initiated by Pawlak (see [112]). Themes in this tradition include the development of modal logics of similarity (or relative similarity) and there are strong links with the tradition of rough-set theory. Konikowska [97] has proposed adding nominals to such logics. Her work is motivated primarily by proof-theoretical considerations: the ability to name states leads to simpler and more intuitive proof systems.

Logics of Space. Nominals have found several applications in modal logics of space. In this chapter, we have treated hybrid languages from a relational perspective, viewing them as language for describing relational structures. Another well known semantics for modal logics is in terms of topological spaces [108]. A topological space is a tuple $\langle X, \Omega\rangle$ where X is a nonempty set and Ω is a collection of subsets of X satisfying three conditions: X and $\emptyset$ are elements

of Ω, every union of elements of Ω is in Ω, and every intersection of finitely many elements of Ω is in Ω. A topological model for the basic modal language, now, consists of a topological space $\langle X, \Omega \rangle$ and a valuation $V : \mathsf{PROP} \to \wp(X)$. The truth definition for modal formulas with respect to such topological models is similar to the one for Kripke models, except that the modal operator $\Box$ is interpreted as follows (where $m \in X$): $\langle X, \Omega, V \rangle, m \models \Box\varphi$ iff $\exists O \in \Omega$ such that $m \in O$ and for all $m' \in O$, $\langle X, \Omega, V \rangle, m' \models \varphi$. This topological semantics is useful for spatial reasoning [19, 1] and modelling knowledge [54]. As in the relational semantics, we can study notions such as validity of a modal formula on a topological space, and modally definable properties of topological spaces. It turns out that, as a language for defining properties of topological spaces, the basic modal language is very weak. In particular, none of the familiar topological separation axioms is modally definable [70].

Nominals can be introduced in topological models in the same way as in Kripke models: they are simply propositional variables whose valuation is always a singleton set. It was noted in [70] that, with the help of nominals, more properties of topological spaces can be defined, including the separation axioms T_0 and T_1. Sustretov [134] has recently proved a topological analogue of Theorem 22, characterizing the properties of topological spaces that can be defined by means of $\mathcal{H}(@)$- and $\mathcal{H}(\mathsf{E})$-formulas. Heinemann [87, 86] has investigated hybrid extensions of the bi-modal logic of knowledge and effort presented in [54], in order to obtain complete axiomatization of frame classes that, while relevant for applications, are not expressible in the basic modal language. In [87], Heinemann provides an axiomatization of the class of linear set spaces, using nominals that denote pairs in $X \times \Omega$. In [86], instead, two sorts of nominals are introduced, ranging over elements of X and Ω, respectively, and topological notions like separation and connectedness are axiomatized.

Nominals have also found applications in logics of metric spaces [101].

Second Order Propositional Modal Logic. In [61], the extension of the basic modal language with propositional quantifiers $\exists p$ and $\forall p$ is studied. This language is called *second order propositional modal logic* (SOPML). It was shown in [136, 137] that there is a close connection between SOPML and $\mathcal{H}(@, \downarrow)$:

THEOREM 45. *Every nominal free $\mathcal{H}(@, \downarrow)$-sentence is equivalent to a formula of SOPML. Conversely, if a formula of SOPML has a first-order equivalent, then it is equivalent to a nominal free $\mathcal{H}(@, \downarrow)$-sentence.*

Theorem 45 shows that, in some sense, nominal-free $\mathcal{H}(@, \downarrow)$ is the intersection of SOPML and first-order logic. This connection was used in [136, 137] to transfer a number of expressivity and frame definability results from $\mathcal{H}(@, \downarrow)$ to SOPML. For example, a first-order formula with one free variable is equivalent to a SOPML-formula iff it is invariant under generated submodels; and an elementary class of frames is definable in SOPML iff it is closed under generated submodels and reflects point-generated subframes (see Theorem 27). For more information about SOPML, see Chapter 10 of this handbook.

Modal Predicate Logics. Nominals can also be added on top of a first-order modal basis (cf. Chapter 9 of this handbook). Blackburn and Marx [29] investigate tableau systems for such first-order hybrid logics, while Braüner [44] discusses natural deduction systems. As in the propositional case, the outcome seems to be a better behaved logical system, that comes with general completeness results.

First-order hybrid logics also have advantages in relation to interpolation and Beth definability. Fine [62] showed that interpolation and the Beth definability property fail for quantified S5 with varying domains, and also for any quantified modal logic between K and S5 with constant

domains. In [9] it is shown that these properties are regained when state variables, satisfaction operators and $\downarrow$ are added to the language. Actually, interpolation and the Beth property hold relative to any bounded fragment definable class of skeletons (the first-order modal analogue of frames), with either varying, expanding, contracting or constant domains. Moreover, the interpolant can be obtained constructively using the techniques of [30].

For further details on first-order hybrid logics, see Chapter 9 in this handbook.

Labeled Deduction. In [68] the notation $l{:}\varphi$ is introduced, where the meta-linguistic symbol : associates the meta-linguistic label l with the object language formula φ. Labeled deduction proceeds by manipulating such expressions, using the labels to guide proof search. Labelled deduction has been successfully used to provide complete and well behaved calculi for a wide range of logics, including non-classical logics where the notion of "state" is usually crucial (see, e.g., [145]). For example, Simpson defines in [132] a family of labeled natural deduction calculi for modal intuitionistic logics and shows that they have good proof theoretic properties; while Kurtonina [99] uses labels to provide complete calculi for categorial type logics, for a variety of frame classes.

One way to see why hybrid languages are proof-theoretically natural, is to observe that nominals and satisfaction operators can capture the main ideas of labeled deduction. Hybrid languages "internalize" labeled deduction into the object language: nominals are essentially object-level labels, and the formula $@_l\varphi$ asserts in the object language what $l{:}\varphi$ asserts in the meta-language. Internalization in the particular case of tableaux is discussed in [25], while the case of sequent calculus is presented in [130]. We have seen examples of such calculi in Section 5. In a recent paper, Braüner and de Paiva discuss similar internalized calculi for hybrid versions of intuitionistic modal logics [45].

Model Checking. In this chapter we take satisfiability and consequence as the main inference problems, but other reasoning tasks are also relevant for many applications.

In [64] Franceschet and de Rijke investigate the following model checking problem for a number of hybrid logics: given a model, or a model and an assignment in case of languages with binders, and a formula φ decide whether there is a state in the model satisfying φ. They provide algorithms for model checking and investigate their complexity. Their main results are summarized in Figure 11, where k is the length of the input formula, n and m are the number of nodes and edges in the model, respectively, and r is the nesting degree of hybrid binders. Names listed as $\mathcal{DH}(\cdot)$ correspond to hybrid extensions of converse propositional dynamic logic. We can see that the presence of binders makes model checking PSPACE-complete (as complex as model checking full first-order logic), and it is, in general, exponential in the nesting level of binders. The paper discusses the impact of these results in applications like querying and constraint evaluation over semistructured data.

In [65], a different kind of model checking is investigated, which is used in formal verification. There, a Kripke structure typically represents a computational system, and paths through the structure denote different possible computations. In *linear time model checking*, formulas are evaluated not on the Kripke structure itself, but on the set of paths through it. Actually, two versions of the linear model checking problem can be defined: the existential linear time model checking problem is to determine whether a given formula is satisfied in some path of the model, while the universal linear time model checking problem asks whether the formula is satisfied in all paths.

Unraveling a Kripke structure into a tree carries some complications in the presence of nominals: if the original structure makes a nominal i true in a state which is involved in a cycle (i.e., the

$\mathcal{H}(\langle R^{-1}\rangle, \mathsf{E}, @)$	$\mathrm{O}(k \cdot (n+m))$
$\mathcal{H}(U, S, \mathsf{E}, @)$	$\mathrm{O}(k \cdot n \cdot m)$
$\mathcal{DH}(\mathsf{E}, @)$	$\mathrm{O}(k \cdot (n+m))$
$\mathcal{H}(\downarrow)$	PSPACE-complete
$\mathcal{H}(\mathsf{E}, @, \downarrow)$	PSPACE-complete
$\mathcal{H}_r(\mathsf{E}, @, \downarrow)$	$\mathrm{O}(k \cdot (n+m) \cdot n^r)$
$\mathcal{DH}_r(\mathsf{E}, @, \downarrow)$	$\mathrm{O}(k \cdot (n+m) \cdot n^r)$
$\mathcal{H}(\exists)$	PSPACE-complete
$\mathcal{H}(\mathsf{E}, @, \exists)$	PSPACE-complete
$\mathcal{H}_r(\mathsf{E}, @, \exists)$	$\mathrm{O}(k \cdot (n+m) \cdot n^{2r})$
$\mathcal{DH}_r(\mathsf{E}, @, \exists)$	$\mathrm{O}(k \cdot (n+m) \cdot n^{2r})$

Figure 11. Complexity of model checking different hybrid languages

state is reachable from itself), the "nominal" i will be true in more than one state after unraveling (actually, the denotation of i will be an infinite set). The authors chose to allow such behavior: the only restriction they make is that nominals denote a single state in the original structure, no other conditions are imposed. Under this interpretation, the complexity of linear time model checking for temporal languages coincides with their hybrid extensions: NP-complete (CONP-complete) for $\mathcal{H}(\langle R^{-1}\rangle, @)$ for existential (universal) linear time model checking, and PSPACE for $\mathcal{H}(U, S, @)$.

The Bounded Fragment. We mentioned the bounded fragment of first-order logic in Section 3.2 and in Theorem 16 we established its tight connection with $\mathcal{H}(@, \downarrow)$.

Bounded formulas have been considered in the literature for a long time. In set theory, where bounded quantifiers are of the form $\exists x.(x \in y \wedge \varphi)$ and $\forall x.(x \in y \rightarrow \varphi)$, the bounded fragment was introduced in 1965 by Levy [102], under the name Δ_0. Δ_0-formulas of set theory have the desirable property of being set-theoretically absolute, meaning that whether a Δ_0-formula $\varphi(x_1, \ldots, x_n)$ holds of sets $a_1, \ldots, a_n$ is independent of the universe of set theory in which $a_1, \ldots, a_n$ reside (cf., for instance, [18]). Bounded formulas have also been considered in the context of arithmetic, where bounded quantifiers are of the form $\exists x.(x \leq y \wedge \varphi)$ and $\forall x.(x \leq y \rightarrow \varphi)$. In fact, there is a field of research of its own called *bounded arithmetic*, which is connected to complexity theory (in particular, to the polynomial hierarchy), propositional proof theory, and the length of propositional proofs [47].

Around 1966, Feferman and Kreisel [60, 59] characterized the bounded fragment as the generated submodel invariant fragment of first-order logic. More precisely, they showed that a first-order formula is equivalent to a bounded formula iff it is invariant under generated submodels. Moreover, it was shown in [59] by means of a cut-free sequent calculus that the bounded fragment has interpolation.

7 DISCUSSION

Hybrid logics form a family of natural extensions of modal logic. The naturalness is confirmed by the fact that nominals have been re-invented on several occasions. Hybrid logics offer two important advantages over modal logics: increased expressive power (e.g., in temporal logic, irreflexivity becomes definable when nominals are added to the language) and a simpler proof

theory (there are many proof systems for hybrid logic, and they come with powerful general completeness results). The general question, then, is:

> How much do we gain by extending our language (e.g., how much extra expressive power), and what price do we pay (e.g., what are the complexity theoretic consequences)?

For a number of hybrid languages, we have explored these question systematically in this chapter. In particular, the expressivity of various hybrid languages has been characterized by means of analogues of the Van Benthem theorem and the Goldblatt-Thomason theorem. Concerning complexity, we saw that nominals and satisfaction operators often do not increase the complexity, although in exceptional cases, adding a single nominal can already cause undecidability. For languages containing the $\downarrow$-binder, on the other hand, undecidability seems to be the rule, rather than the exception.

We would like to close this chapter by observing that "hybridization", as an operation on logical languages, can be applied in many contexts. As we discussed, nominals have a natural interpretation not only in the relational (Kripke) semantics, but also in topological and algebraic semantics. The hybrid machinery can be added to the basic modal language, or on top of first-order or higher-order modal languages, and with either a classical or an intuitionistic base. Some of these combinations have been investigated (for instance, several recent papers study topological modal language containing nominals), other remain to be explored.

ACKNOWLEDGEMENTS

We would like to express our gratitude to Ulrike Sattler for her detailed comments and suggestions as second reader for this chapter, and also to Patrick Blackburn who read through earlier versions of the chapter and gave us valuable feedback. Balder ten Cate is supported by NWO grants 612.069.006 and 639.021.508.

BIBLIOGRAPHY

[1] M. Aiello and J. van Benthem. Logical Patterns in Space. In D. Barker-Plummer, D. Beaver, J. van Benthem, and P. Scotto di Luzio, editors, *Words, Proofs, and Diagrams*, pages 5–25. CSLI, Stanford, 2002.

[2] J. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, July 1984.

[3] C. Areces. HyLo: The hybrid logics' web site. `http://hylo.loria.fr`. Last visited: Dec/05.

[4] C. Areces. *Logic Engineering. The Case of Description and Hybrid Logics*. PhD thesis, University of Amsterdam, October 2000. ILLC Dissertation Series DS-2000-05.

[5] C. Areces and P. Blackburn, editors. *Methods for Modalities 3*, Nancy, France, 2003.

[6] C. Areces, P. Blackburn, and M. Marx. A road-map on complexity for hybrid logics. In J. Flum and M. Rodríguez-Artalejo, editors, *Computer Science Logic. Proceedings of the 8th Annual Conference of the EACSL*, number 1683 in LNCS, pages 307–321. Springer, 1999.

[7] C. Areces, P. Blackburn, and M. Marx. The computational complexity of hybrid temporal logics. *Logic Journal of the IGPL*, 8(5):653–679, 2000.

[8] C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation and complexity. *Journal of Symbolic Logic*, 66(3):977–1010, 2001.

[9] C. Areces, P. Blackburn, and M. Marx. Repairing the interpolation lemma in quantified modal logic. *Annals of Pure and Applied Logics*, 123(1–3):287–299, 2003.

[10] C. Areces, H. de Nivelle, and M. de Rijke. Prefixed resolution: a resolution method for modal and description logics. In *Automated Deduction—CADE-16 (Trento, 1999)*, pages 187–201. Springer, Berlin, 1999.

[11] C. Areces, H. de Nivelle, and M. de Rijke. Resolution in modal, description and hybrid logic. *Journal of Logic and Computation*, 11(5):717–736, 2001.

[12] C. Areces and M. de Rijke. From description to hybrid logics, and back. In F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev, editors, *Advances in Modal Logic. Volume 3*, pages 17–36. CSLI Publications, 2001.

[13] C. Areces, E. Franconi, R. Goré, M. de Rijke, and H. Schlingloff, editors. *Methods for Modalities 1*, 2000. Appeared as a Special Issue of the Logic Journal of the IGPL 8(3).

[14] C. Areces and D. Gorín. Ordered resolution with selection for h(@). In F. Baader and A. Voronkov, editors, *Proceedings of LPAR 2004*, volume 3452 of *LNCS*, pages 125–141, Montevideo, Uruguay, 2005. Springer.

[15] C. Areces, J. Heguiabehere, and D. Gorín. HyLoRes: A hybrid logic resolution system. `http://www.loria.fr/~areces/HyLoRes`. Last visited: Dec/05.

[16] F. Baader, D. McGuiness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, implementation and applications*. Cambridge University Press, 2003.

[17] L. Bachmair and H. Ganzinger. Resolution theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning, Volume 1*, chapter 2, pages 19–99. Elsevier Science, 2001.

[18] J. Barwise. *Admissible Sets and Structures*. Springer-Verlag, 1975.

[19] B. Bennett. Modal logics for qualitative spatial reasoning. *Logic Journal of the IGPL*, 4(1), 1996.

[20] N. Bezhanishvili and B. ten Cate. Transfer results for hybrid logic. Part I: the case without satisfaction operators. *Journal of Logic and Computation*, 16(2):177–197, 2006.

[21] P. Blackburn. *Nominal Tense Logic and Other Sorted Intensional Frameworks*. PhD thesis, Centre for Cognitive Science, University of Edinburgh, Edinburgh, United Kingdom, 1990.

[22] P. Blackburn. Nominal tense logic. *Notre Dame Journal of Formal Logic*, 34(1):56–83, 1992.

[23] P. Blackburn. Modal logic and attribute value structures. In M. de Rijke, editor, *Diamonds and Defaults*, Synthese Language Library, pages 19–65. Kluwer Academic Publishers, Dordrecht, 1993.

[24] P. Blackburn. Tense, temporal reference and tense logic. *Journal of Semantics*, 11:83–101, 1994.

[25] P. Blackburn. Internalizing labelled deduction. *Journal of Logic and Computation*, 10:136–168, 2000.

[26] P. Blackburn. Representation, reasoning, and relational structures: a Hybrid Logic manifesto. In Areces et al. [13], pages 339–365. Appeared as a Special Issue of the Logic Journal of the IGPL 8(3).

[27] P. Blackburn. Modal logic as dialogical logic. *Synthese*, 127:57–93, 2001.

[28] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[29] P. Blackburn and M. Marx. Tableaux for quantified hybrid logic. In U. Egly and C. Fermüller, editors, *Automated reasoning with analytic tableaux and related methods, TABLEAUX 2002*, volume 2381 of *LNAI*, pages 38–52. Springer Verlag, 2002.

[30] P. Blackburn and M. Marx. Constructive interpolation in hybrid logic. *The Journal of Symbolic Logic*, 68(2):463–479, 2003.

[31] P. Blackburn and J. Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4(3):251–272, 1995. Special issue on decompositions of first-order logic.

[32] P. Blackburn and J. Seligman. What are hybrid languages? In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic 1*, pages 41–62. CSLI Publications, Stanford University, 1998.

[33] P. Blackburn and E. Spaan. A modal perspective on the computational complexity of attribute value grammar. *Journal of Logic, Language and Information*, 2:129–169, 1993.

[34] P. Blackburn and B. ten Cate. Pure extensions, proof rules, and hybrid axiomatics. In Schmidt et al. [127], pages 19–29.

[35] P. Blackburn and M. Tzakova. Hybrid completeness. *Logic Journal of the IGPL*, 6:625–650, 1998.

[36] P. Blackburn and M. Tzakova. Hybridizing concept languages. *Annals of Mathematics and Artificial Intelligence*, 24:23–49, 1998.

[37] P. Blackburn and M. Tzakova. Hybrid languages and temporal logic. *Logic Journal of the IGPL*, 7(1):27–54, 1999.

[38] T. Bolander and T. Braüner. Two tableau-based decision procedures for hybrid logic. In Schlinglof [126].

[39] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Springer-Verlag, Berlin, 1997. With an appendix by C. Allauzen and B. Durand.

[40] A. Borgida and P. Patel-Schneider. A semantics and complete algorithm for subsumption in the CLASSIC description logic. *Journal of Artificial Intelligence Research*, 1:277–308, 1994.

[41] R. Brachman and J. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[42] T. Braüner. Natural deduction for hybrid logic. *Journal of Logic and Computation*, 14(3):329–353, 2004.

[43] T. Braüner. Two natural deduction systems for hybrid logic: A comparison. *Journal of Logic, Language and Information*, 13(1):1–23, 2004.

[44] T. Braüner. Natural deduction for first-order hybrid logic. *Journal of Logic, Language and Information*, 14(2):173 – 198, 2005.

[45] T. Braüner and V. de Paiva. Towards constructive hybrid logic. In Areces and Blackburn [5].

[46] R. Bull. An approach to tense logic. *Theoria*, 36:282–300, 1970.

[47] S. Buss. Bounded arithmetic and propositional proof complexity. In H. Schwichtenberg, editor, *Logic and Computation*, pages 67–122. Springer-Verlag, Berlin, 1997.

[48] D. Calvanese, G. De Giacomo, and M. Lenzerini. Reasoning in expressive description logics with fixpoints based on automata on infinite trees. In T. Dean, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 84–89, Stockholm, Sweden, 1999. Morgan-Kaufmann Publishers.

[49] A. Chagrov and M. Zakharyaschev. *Modal logic*. Oxford University Press, 1997.

[50] C. Chen and I. Lin. The complexity of propositional modal theories and the complexity of consistency of propositional modal theories. In A. Nerode and Y. Matiyasevich, editors, *Proceedings of the Third International Symposium on Logical Foundations of Computer Science*, volume 813 of *Lecture Notes in Computer Science*, pages 69–80. Springer, 1994.

[51] B. Copeland. Arthur Prior. In E. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 1999. `http://plato.stanford.edu/archives/win1999/entries/prior`.

[52] M. Cresswell. *Entities and Indices*. Reidel, Dordrecht, 1990.

[53] M. Cresswell. *Semantic Indexicality*. Kluwer, Dordrecht, 1996.

[54] A. Dabrowski, L. Moss, and R. Parikh. Topological reasoning and the logic of knowledge. *Annals of Pure and Applied Logic*, 78:73–110, 1996.

[55] G. De Giacomo. *Decidability of class-based knowledge representation formalisms*. PhD thesis, Universitá di Roma 'La Sapienza", 1995.

[56] G. De Giacomo and M. Lenzerini. Boosting the correspondence between description logics and propositional dynamic logics. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI'94)*, pages 205–212, 1994.

[57] S. Demri. Sequent calculi for nominal tense logics: a step towards mechanization? In N. Murray, editor, *Conference on Tableaux Calculi and Related Methods (TABLEAUX), Saratoga Springs, USA*, volume 1617 of *LNAI*, pages 140–154. Springer Verlag, 1999.

[58] H. Enderton. *A mathematical introduction to logic*. Academic Press, New York, 1972.

[59] S. Feferman. Persistent and invariant formulas for outer extensions. *Compositio Mathematica*, 20:29–52, 1968.

[60] S. Feferman and G. Kreisel. Persistent and invariant formulas relative to theories of higher order. *Bulletin of the American Mathematical Society*, 72:480–485, 1966. Research Announcement.

[61] K. Fine. Propositional quantifiers in modal logic. *Theoria*, 36:336–346, 1970.

[62] K. Fine. Failures of the interpolation lemma in quantified modal logic. *Journal of Symbolic Logic*, 44(2):201–206, 1979.

[63] M. Fitting. *First order logic and automated theorem proving*. Springer Verlag, 2 edition, 1996.

[64] M. Franceschet and M. de Rijke. Model checking for hybrid logics. In Areces and Blackburn [5], pages 109–124.

[65] M. Franceschet, M. de Rijke, and H. Schlingloff. Hybrid logics on linear structures: Expressivity and complexity. In *Proceedings of the 10th International Symposium on Temporal Representation and Reasoning / 4th International Conference on Temporal Logic (TIME-ICTL 2003)*, pages 166–173, Cairns, Queensland, Australia, July 2003. IEEE Computer Society.

[66] D. Gabbay. *Investigations in Modal and Tense Logics with Applications to Problems in Philosophy and Linguistics*. Reidel, Dordrecht, 1976.

[67] D. Gabbay. An irreflexivity lemma. In U. Mönnich, editor, *Aspects of Philosophical Logic*, pages 67–89. Reidel, 1981.

[68] D. Gabbay. *Labelled Deductive Systems. Vol. 1.* The Clarendon Press Oxford University Press, New York, 1996. Oxford Science Publications.

[69] D. Gabbay, Á. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*, volume 148 of *Studies in Logic*. North-Holland, 2003.

[70] D. Gabelaia. Modal definability in topology. Master's thesis, University of Amsterdam, 2001. ILLC MSc thesis series MoL-2001-11.

[71] G. Gargov. Decidability of the basic combinatory propositional dynamic logic (in Russian). In A. Ershov and D. Skordev, editors, *Mathematical Theory of Programming*, pages 42–49. Computer Centre of the Siberian Division of Soviet Academy of Sciences, Novosibirsk, 1985.

[72] G. Gargov and V. Goranko. Modal logic with names. *Journal of Philosophical Logic*, 22(6):607–636, 1993.

[73] G. Gargov and S. Passy. Determinism and looping in combinatory PDL. *Theoretical Computer Science*, 61(2-3):259–277, 1988.

[74] G. Gargov, S. Passy, and T. Tinchev. Modal environment for boolean speculations. In D. Scordev, editor, *Mathematical Logic and its Applications*, pages 253–263. Plenum Press, New York, 1987.

[75] R. Goldblatt. *Axiomatizing the Logic of Computer Programming*. Number 130 in LNCS. Springer, Berlin, 1982.

[76] R. Goldblatt and S. Thomason. Axiomatic classes in propositional modal logic. In J. Crossley, editor, *Algebra and Logic*, pages 163–173. Springer, 1974.

[77] V. Goranko. Temporal logic with reference pointers. In *Temporal Logic*, volume 827 of *Lecture Notes in Artificial Intelligence*, pages 133–148. Springer-Verlag, 1994.

[78] V. Goranko. Hierarchies of modal and temporal logics with reference pointers. *Journal of Logic, Language and Information*, 5(1):1–24, 1996.

[79] V. Goranko. Computation tree logics and temporal logics with reference pointers. *Journal of Applied Non-classical Logics*, 10(3-4):221–242, 2000.

[80] V. Goranko and D. Vakarelov. Sahlqvist formulae in hybrid polyadic modal logics. *Journal of Logic and Computation*, 11(5):737–754, 2001.

[81] R. Goré, A. Leitsch, and T. Nipkow, editors. *Proceedings of the 5th Internaltion Joint Conference on Automated Reasoning (IJCAR'01)*, number 2083 in Lecture Notes in Artificial Intelligence. Springer-Verlag, 2001.

[82] Erich Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64(4):1719–1742, 1999.
[83] V. Haarslev and R. Möller. RACER: Renamed ABox and concept expression reasoner. `http://www.sts.tu-harburg.de/~r.f.moeller/racer`. Last visited: Dec/05.
[84] V. Haarslev and R. Möller. RACER system description. In Goré et al. [81], pages 701–705.
[85] D. Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic. Vol. II*, volume 165 of *Synthese Library*, pages 497–604. D. Reidel Publishing Co., Dordrecht, 1984. Extensions of classical logic.
[86] B. Heinemann. A hybrid logic of knowledge supporting topological reasoning. In C. Rattray, S. Maharaj, and C. Shankland, editors, *Algebraic Methodology and Software Technology, AMAST 2004*, volume 3116 of *Lecture Notes in Computer Science*, pages 181–195, Berlin, 2004. Springer.
[87] B. Heinemann. The hybrid logic of linear set spaces. *Logic Journal of the IGPL*, 12(3):181–198, 2004.
[88] T. Henzinger. Half-order modal logic: how to prove real-time properties. In *PODC '90: Proceedings of the 9th annual ACM Symposium on Principles of Distributed Computing*, pages 281–296. ACM Press, 1990.
[89] I. Horrocks. FaCT: Fast classification of terminologies. `http://www.cs.man.ac.uk/~horrocks/FaCT`. Last visited: Dec/05.
[90] I. Horrocks. Using an expressive description logic: FaCT or fiction? In *Proceedings of the 6th International Conference on the Principles of Knowledge Representation and Reasoning (KR98)*, pages 636–647, 1998.
[91] I. Horrocks and U. Sattler. Ontology reasoning in the $\mathcal{SHOQ}(D)$ description logic. In B. Nebel, editor, *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 199–204, Seattle, Washington, USA, August 2001. Morgan-Kaufmann Publishers.
[92] I. Horrocks and U. Sattler. A tableaux decision procedure for $\mathcal{SHOIQ}$. In L. Pack Kaelbling and A. Saffiotti, editors, *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 448–453, Edingurgh, UK, August 2005. Professional Book Center.
[93] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. In Areces et al. [13], pages 239–264.
[94] I. Horrocks, U. Sattler, and S. Tobies. Reasoning with individuals for the description logic $\mathcal{SHIQ}$. In D. MacAllester, editor, *Proceedings of the 17th International Conference on Automated Deduction (CADE-17)*, Lecture Notes in Computer Science, pages 482–496, Germany, 2000. Springer Verlag.
[95] A. Hussain. A minimal hybrid logic for intervals. *Logic Journal of the IGPL*, 14(1):35–62, 2006.
[96] H. Kamp. Formal properties of "now". *Theoria*, 37:227–273, 1971.
[97] B. Konikowska. A logic for reasoning about relative similarity. *Studia Logica*, 58:185–226, 1997.
[98] B. Kooi, G. Renardel de Lavalette, and R. Verbrugge. Strong completeness for non-compact hybrid logics. In Schmidt et al. [127], pages 212–223.
[99] N. Kurtonina. *Frame and Labels*. PhD thesis, Utrecht University, 1994.
[100] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of abstract description systems. *Artificial Intelligence*, 156(1):1–73, 2004.
[101] O. Kutz, F. Wolter, H. Sturm, N. Suzuki, and M. Zakharyaschev. Logics of metric spaces. *ACM Transactions on Computational Logic*, 4(2):260–294, 2003.
[102] A. Lévy. *A hierarchy of formulas in set theory*, volume 57 of *Memoirs of the American Mathematical Society*. American Mathematical Society, 1965.
[103] C. Lutz. An improved NExpTime-hardness result for the description logic $\mathcal{ALC}$ extended with inverse roles, nominals, and counting. LTCS-Report LTCS-04-07, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany, 2004.
[104] R. MacGregor. Inside the LOOM description classifier. *SIGART Bulletin*, 2(3):88–92, 1991.
[105] M. Marx. Narcissists, stepmothers and spies. In G. Görz, V. Haarslev, C. Lutz, and R. Möller, editors, *Proceedings of the 2002 International Workshop on Description Logics (DL'02)*, Toulouse, France, 2002.
[106] M. Marx and Y. Venema. *Multidimensional Modal Logic*, volume 4 of *Applied Logic Series*. Kluwer Academic Publishers, Dordrecht, 1997.
[107] D. McGuinness and F. van Harmelen. OWL Web Ontology Language. Overview. `http://www.w3.org/TR/owl-features`. Last visited: Dec/05.
[108] J. McKinsey and A. Tarski. The algebra of topology. *Annals of Mathematics*, 45:141–191, 1944.
[109] J. McTaggart. The unreality of time. *Mind*, pages 457–474, 1908.
[110] M. Mundhenk, T. Schneider, T. Schwentick, and V. Weber. Complexity of hybrid logics over transitive frames. In Schlinglof [126].
[111] P. Øhrstrøm and P. Hasle. A. N. Prior's rediscovery of tense logic. *Erkenntnis*, 39:23–50, 1993.
[112] E. Orłowska, editor. *Incomplete Information: Rough Set Analysis*. Studies in Fuzziness and Soft Computing. Physica-Verlag, 1997.
[113] S. Passy and T. Tinchev. PDL with data constants. *Information Processing Letters*, 20(1):35–41, 1985.
[114] S. Passy and T. Tinchev. Quantifiers in combinatory PDL: completeness, definability, incompleteness. In *Fundamentals of Computation Theory FCT 85*, volume 199 of *LNCS*, pages 512–519. Springer, 1985.
[115] S. Passy and T. Tinchev. An essay in combinatory dynamic logic. *Information and Computation*, 93(2):263–332, 1991.
[116] A. Prior. The syntax of time-distinctions. *Franciscan Studies*, 18:105–120, 1958.
[117] A. Prior. *Past, Present and Future*. Clarendon Press, Oxford, 1967.

[118] A. Prior. *Papers on Time and Tense*. University of Oxford Press, 1968.

[119] A. Prior. *Worlds, Times and Selves*. Duckworth, London, 1977. Edited by K. Fine.

[120] M. Reape. A feature value logic. In C. Rupp, M. Rosner, and R. Johnson, editors, *Constraints, Language and Computation*, Synthese Language Library, pages 77–110. Academic Press, 1994.

[121] M. Reynolds. An axiomatization of full computation tree logic. *Journal of Symbolic Logic*, 66(3):1011–1057, 2001.

[122] B. Richards, I. Bethke, J. van der Does, and J. Oberlander. *Temporal Representation and Inference*. Academic Press, London, 1989.

[123] W. Rounds. Feature logics. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 475–533. North-Holland Publishing Co., Amsterdam, 1997.

[124] U. Sattler and M. Vardi. The hybrid μ-calculus. In Goré et al. [81], pages 76–91.

[125] K. Schild. A correspondence theory for terminological logics. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI'91)*, pages 466–471, Sydney, Australia, August 1991. Morgan-Kaufmann Publishers.

[126] H. Schlinglof, editor. *Methods for Modalities 4*, Berlin Adlershof, Germany, December 2005.

[127] R. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing, editors. *Proceedings of 5th AiML Conference*, Manchester, UK, 2004. Department of Computer Science, University of Manchester.

[128] J. Seligman. A cut-free sequent calculus for elementary situated reasoning. Technical Report HCRC/RP-22, HCRC, Edinburgh, 1991.

[129] J. Seligman. The logic of correct description. In M. de Rijke, editor, *Advances in Intensional Logic*, pages 107–135. Kluwer, 1997.

[130] J. Seligman. Internalization: The case of hybrid logics. *Journal of Logic and Computation*, 11(5):671–689, 2001. Special Issue on Hybrid Logics. Areces, C. and Blackburn, P. (eds.).

[131] M. Sellink. Verifying modal formulae over I/O-automata by means of type theory. Logic group preprint series, Utrecht University, 1994.

[132] A. Simpson. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. PhD thesis, University of Edinburgh, 1994.

[133] E. Spaan. *Complexity of Modal Logics*. PhD thesis, ILLC, University of Amsterdam, 1993.

[134] D. Sustretov. Hybrid definability in topological spaces. In J. Gervain, editor, *Proceedings of ESSLLI'2005 Student Session*, pages 342–352, 2005.

[135] B. ten Cate. Interpolation for extended modal languages. *Journal of Symbolic Logic*, 70(1):223–234, 2005.

[136] B. ten Cate. *Model theory for extended modal languages*. PhD thesis, University of Amsterdam, 2005. ILLC Dissertation Series DS-2005-01.

[137] B. ten Cate. Expressivity of second-order propositional modal logic. *Journal of Philosophical Logic*, 35: 209–223, 2006

[138] B. ten Cate and M. Franceschet. Guarded fragments with constants. *Journal of Logic, Language and Information*, 14(3):281–288, 2005.

[139] B. ten Cate and M. Franceschet. On the complexity of hybrid logics with binders. In L. Ong, editor, *Proceedings of 5th AiML Conference*, volume 3634 of *Lecture Notes in Computer Science*, pages 339–354. Springer Verlag, 2005.

[140] B. ten Cate, M. Marx, and P. Viana. Hybrid logics with Sahlqvist axioms. *Logic Journal of the IGPL*, 13(3):293–300, 2005.

[141] M. Tzakova. *Hybrid Languages*. PhD thesis, Technischen Fakultät der Universität des Saarlandes, 1999.

[142] M. Tzakova. Tableaux calculi for hybrid logics. In N. Murray, editor, *Proceedings of the Conference on Tableaux Calculi and Related Methods (TABLEAUX), Saratoga Springs, USA*, volume 1617 of *LNAI*, pages 278–292. Springer Verlag, 1999.

[143] M. Vardi. Reasoning about the past with two-way automata. In K. Larsen, S. Skyum, and G. Winskel, editors, *Proceedings of the 25th International Colloquium on Automata, Languages and Programming*, Lecture Notes In Computer Science, pages 628–641. Springer-Verlag, 1998.

[144] Y. Venema. *Many-dimensional modal logic*. PhD thesis, ILLC, University of Amsterdam, 1991.

[145] L. Viganò. *Labelled Non-Classical Logics*. Kluwer Academic Publishers, 2000.

[146] F. Vlach. *Now and Then: a formal study in the logic of tense anaphora*. PhD thesis, UCLA, 1973.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

15

COMBINING MODAL LOGICS

Agi Kurucz

1 Introduction . . . 869
2 Fusion of modal logics . . . 872
2.1 Transfer results . . . 872
2.2 Complexity of fusions . . . 874
3 Product of modal logics . . . 875
3.1 General transfer results . . . 878
3.2 Connections with other formalisms . . . 881
3.3 Axiomatising products . . . 884
3.4 Decision problems and complexity of products . . . 889
4 Between fusions and products . . . 910
5 Other combinations . . . 917

1 INTRODUCTION

When can we say that a logic is a *combination* of others? In general, *any* logical system having more than one connective can be considered as a combination of logical systems having fewer connectives. In particular, *any* multimodal logic can be considered as a combination of, say, unimodal logics. So, in this general sense, *any* result on multimodal logics can be considered as a result on combining modal logics. What makes this chapter special among other ones studying multimodal logics is that here we investigate the following kind of problems:

> *Given a family* L *of modal logics and a combination method* C*, do certain properties of the 'component logics'* $L \in \mathsf{L}$ *transfer to their 'combination'* $\mathsf{C}(\mathsf{L})$*?*

Most of the *combination methods* considered in this chapter satisfy the following three criteria:

(C1) They are *finitary*, that is, C is defined only on *finite* families L of modal logics.

(C2) The combination $\mathsf{C}(\mathsf{L})$ of (multi)modal logics from L is a (multi)modal logic itself.

(C3) The combined logic $\mathsf{C}(\mathsf{L})$ is an extension of each component logic $L \in \mathsf{L}$.

For each considered combination method, we discuss in detail the possible transfer of the following two kinds of properties:

- **Axiomatisation/completeness.**
  There are two versions, depending on whether the combination method results in a syntactically or semantically defined logic. In the former case, the question is whether the combination of recursively (finitely) axiomatisable components remains recursively (finitely) axiomatisable, and in the latter, whether the Kripke completeness of the components transfers to their combination.

- **Decidability/complexity of the validity/satisfiability problem.**
  We study whether decidability of the validity problem transfers from the components to their combination and if so, what is the change in complexity. We also discuss the possible transfer of the finite model property.

For transfer results about several other properties (like versions of interpolation, decidability of various consequence relations, etc.) see [23] and the references therein. Combinations of deductive calculi (such as combined tableaux) are not considered either, see Chapter 2 of this handbook for some examples.

Combination methods not satisfying **(C1)–(C3)** are in general out of our scope, though see Section 5 for a discussion.

Notation and terminology. We will mainly consider possible world (or Kripke) semantics. *Kripke models* are pairs $\mathfrak{M} = \langle \mathfrak{F}, \mathfrak{V} \rangle$ that are based on relational structures $\mathfrak{F} = \langle W, R_1, \ldots, R_n \rangle$, where $n > 0$ is a natural number, W is a non-empty set and the R_i are binary relations on it. Such structures are called *n-frames* (or *frames*, for short). We say that an n-frame $\mathfrak{G} = \langle U, S_1, \ldots, S_n \rangle$ is a *subframe* of an n-frame $\mathfrak{F}$ ($\mathfrak{G} \subseteq \mathfrak{F}$, in symbols) if $U \subseteq W$ and $S_i = R_i \cap (U \times U)$, for $i = 1, \ldots, n$. A *path of length k* from point x to point y in an n-frame $\mathfrak{F}$ is a sequence $\langle x_0, \ldots, x_k \rangle$ of points such that $x_0 = x$, $x_k = y$ and $x_i R_j x_{i+1}$, for each $i < k$ and some j, $1 \le j \le n$. We call an n-frame $\mathfrak{F}$ *rooted* if there exists some $x \in W$ such that for every $y \in W$, $y \ne x$ there is a path from x to y. Such an x is called a *root of* $\mathfrak{F}$. We say that $\mathfrak{F}$ is of *depth k* if k is the length of the longest path in $\mathfrak{F}$. If such a longest path does not exist, then we say that $\mathfrak{F}$ is of *infinite depth*. An n-frame $\mathfrak{F}$ is called *tree-like* if it is rooted and $R = \bigcup_{i=1}^{n} R_i$ is weakly connected on the set $\{y \in W \mid yRx\}$ for every $x \in W$. If a tree-like frame is well-founded (i.e., there are no infinite descending R-chains $\ldots Rx_2Rx_1Rx_0$ of points) then we call $\mathfrak{F}$ a *tree*. The *depth* $d^{\mathfrak{F}}(x)$ of a point x in a tree $\mathfrak{F}$ is defined to be the length of the unique path from the root to x. If for no $n < \omega$ the point x is of depth n, then we say that x is of *infinite depth*. By the *co-depth* of a point x in a tree $\mathfrak{F}$ we understand the depth of the subtree of $\mathfrak{F}$ with root x.

Given a natural number n, the *n-modal language* $\mathcal{ML}_n$ has propositional variables $p, q, s, \ldots$, Boolean connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\top$, $\bot$, and (unary) modal operators $\Box_1, \ldots, \Box_n$ and $\Diamond_1, \ldots, \Diamond_n$. $\mathcal{ML}_n$-formulas are formed inductively in the usual way. Given an $\mathcal{ML}_n$-formula φ, we let $sub\,\varphi$ denote the set of all subformulas of φ, and $md(\varphi)$ denote the *modal depth* of φ. We will also use the following abbreviations. For every formula φ, let

$$\Box^0\varphi = \varphi \qquad \text{and, for } n < \omega, \quad \Box^{n+1}\varphi = \Box\Box^n\varphi, \quad \Box^{\le n}\varphi = \bigwedge_{k \le n} \Box^k\varphi.$$

The *truth-relation* '$(\mathfrak{M}, w) \models \varphi$' connecting syntax and semantics is defined by induction on the construction of φ as usual. We say that φ is *true in* $\mathfrak{M}$ ($\mathfrak{M} \models \varphi$, in symbols),

if $\mathfrak{M}, w \models \varphi$ for all $x \in W$. A formula φ is said to be *valid in a frame* $\mathfrak{F}$ ($\mathfrak{F} \models \varphi$, in symbols), if $\mathfrak{M} \models \varphi$ for every model $\mathfrak{M}$ that is based on $\mathfrak{F}$. Given a set Σ of formulas, we set

$$\mathsf{Fr}\,\Sigma \;=\; \{\mathfrak{F} \mid \mathfrak{F} \models \varphi,\ \text{for all } \varphi \in \Sigma\}.$$

If $\mathfrak{M} \models \varphi$ for all $\varphi \in \Sigma$ then we say that $\mathfrak{M}$ is a *model for* Σ. Similarly, φ is said to be a *frame for* Σ, if $\mathfrak{F} \in \mathsf{Fr}\,\Sigma$.

By an *n-modal logic* (or *modal logic*[1], for short) we mean any set L of $\mathcal{ML}_n$-formulas that contains all valid formulas of classical propositional logic, the formulas

$$\text{(K)} \qquad \Box_i(p \to q) \to (\Box_i p \to \Box_i q),$$

and is closed under the rules of Substitution, Modus Ponens and Necessitation, for $i = 1, \dots, n$ (see Chapter 2 of this handbook).

Let us briefly discuss two of the most common ways of defining a modal logic: the 'syntactical' way (via axioms) and the 'semantical' way (via a class of intended frames). First, given a set Σ of $\mathcal{ML}_n$-formulas and an n-modal logic L, we say that L *is axiomatised by* Σ, if L is the smallest n-modal logic containing Σ. If Σ can be chosen a recursive (or finite) subset of all $\mathcal{ML}_n$-formulas, then we say that L is *recursively* (*finitely*) *axiomatisable*. And second, given a class $\mathcal{C}$ of n-frames, the set

$$\mathsf{Log}\,\mathcal{C} = \{\varphi \mid \mathfrak{F} \models \varphi,\ \text{for all } \mathfrak{F} \in \mathcal{C}\}$$

is always an n-modal logic. An n-modal logic L is called *Kripke complete* if $L = \mathsf{Log}\,\mathcal{C}$ for some class $\mathcal{C}$ of n-frames. In this case we also say that L is *characterised* (or *determined*) by $\mathcal{C}$. As is well-known, there exist incomplete modal logics, and similarly, there are Kripke complete logics that are not recursively axiomatisable (see Section 3.4 for some examples).

The *validity problem* for an n-modal logic L is the problem of deciding whether a given $\mathcal{ML}_n$-formula belongs to L or not. If this problem is decidable (or recursively enumerable) then we also say that the logic L is *decidable* (or *recursively enumerable*). A related problem is the *satisfiability problem* for L: given φ, decide whether φ is *L-satisfiable*, that is, whether there exists a model $\mathfrak{M}$ for L and a world w in $\mathfrak{M}$ such that $\mathfrak{M}, w \models \varphi$ holds. It is easy to see the connection between the two: $\varphi \in L$ iff $\neg\varphi$ is not L-satisfiable. Given a recursively enumerable logic L, we can have a decision algorithm for L if we can enumerate those formulas that are not in L. Clearly, this can be done if:

- the class of finite frames for L is recursively enumerable (up to isomorphism, of course), and
- L has the *finite model property*[2], that is

$$L = \mathsf{Log}\,\{\mathfrak{F} \in \mathsf{Fr}\,L \mid \mathfrak{F} \text{ is finite}\}.$$

This chapter is not self-contained in the sense that we discuss well-known modal logics like **K**, **S5**, **KD45**, **K4**, **S4**, **K4.3**, **GL**, **Grz**, **Alt**, etc. without defining them. We also use without explicit reference standard notions and results from basic modal logic,

[1] We consider here what are usually called *normal* modal logics only.

[2] It would be more precise to call this *finite frame property*. However, as is well-known, it is equivalent to saying that $L = \{\varphi \mid \mathfrak{M} \models \varphi,\ \text{for all finite models } \mathfrak{M} \text{ for } L\}$.

such as p-morphisms and disjoint unions, generated subframes, unravelling, results on Sahlqvist formulas and canonicity, etc. For notions and statements not defined or proved here, see other chapters in this handbook or [12, 10].

2 FUSION OF MODAL LOGICS

Within the constraints **(C1)**–**(C3)** above, the formation of *fusions* (also known as *independent joins*), is the simplest and perhaps the most natural way of combining modal logics:

DEFINITION 1. Let L_1 and L_2 be two modal logics formulated in languages $\mathcal{ML}_n$ and $\mathcal{ML}_m$ in such a way that they have disjoint sets of modal operators (say, $\Box_1, \dots, \Box_n$ and $\Box_{n+1}, \dots, \Box_{n+m}$, respectively). Then the *fusion*

$$L_1 \otimes L_2$$

of L_1 *and* L_2 is the smallest $(n+m)$-modal logic L containing both L_1 and L_2.

It is easy to see that if each L_i is axiomatised by a set Σ_i of axioms (written in the respective languages) then $L_1 \otimes L_2$ is axiomatised by the union $\Sigma_1 \cup \Sigma_2$. This means that *no* axiom containing modal operators from both of the languages of L_1 and L_2 is required to axiomatise $L_1 \otimes L_2$. In other words, in fusions the modal operators of the component logics are kind of 'independent,' they 'do not interact'.

The formation of fusions is clearly an associative binary operation on modal logics. Therefore, one can define the fusion

$$L_1 \otimes L_2 \otimes \cdots \otimes L_n$$

of n modal logics in a straightforward way, for any natural number $n \geq 2$. Observe that well-known multimodal logics like $\mathbf{K}_n$ or $\mathbf{S5}_n$ are the fusions of their unimodal 'counterparts':

$$\mathbf{K}_n = \underbrace{\mathbf{K} \otimes \cdots \otimes \mathbf{K}}_{n}, \qquad \mathbf{S5}_n = \underbrace{\mathbf{S5} \otimes \cdots \otimes \mathbf{S5}}_{n}.$$

The formation of fusions as a combination method does satisfy criterion **(C3)**, as the following result of Thomason [78] shows:

THEOREM 2. *The fusion of consistent modal logics is a conservative extension of the components.*

2.1 Transfer results

We begin with the following result of Kracht and Wolter [48], and Fine and Schurz [16] stating that *Kripke completeness* of the components transfers to their fusion:

THEOREM 3. *If modal logics* L_1 *and* L_2 *are characterised by classes of frames* $\mathcal{C}_1$ *and* $\mathcal{C}_2$*, respectively, and if* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *are closed under the formation of disjoint unions and isomorphic copies, then the fusion* $L_1 \otimes L_2$ *of* L_1 *and* L_2 *is characterised by the class*

$$\mathcal{C}_1 \otimes \mathcal{C}_2 = \big\{ \langle W, R_1, \dots, R_n, S_1, \dots, S_m \rangle \mid \langle W, R_1, \dots, R_n \rangle \in \mathcal{C}_1,\ \langle W, S_1, \dots, S_m \rangle \in \mathcal{C}_2 \big\}.$$

It should be clear that if $\mathcal{C}_1$ and $\mathcal{C}_2$ determine logics L_1 and L_2, respectively, then all frames in $\mathcal{C}_1 \otimes \mathcal{C}_2$ are frames for the fusion $L_1 \otimes L_2$. Let us outline the proof of the converse statement, i.e., that $\mathcal{C}_1 \otimes \mathcal{C}_2$ actually characterises $L_1 \otimes L_2$. To simplify notation, we assume that L_1 and L_2 are unimodal logics with the boxes $\Box_1$ and $\Box_2$, respectively. The fusion $L = L_1 \otimes L_2$ is then a bimodal logic in the language $\mathcal{ML}_2$.

With each $\mathcal{ML}_2$-formula φ of the form $\Box_i\psi$ $(i = 1, 2)$ we associate a new variable q_φ which will be called the *surrogate* of φ. For an $\mathcal{ML}_2$-formula φ containing no surrogate variables, denote by $\varphi^{\mathbf{1}}$ the formula that results from φ by replacing all its subformulas of the form $\Box_2\psi$, which are not within the scope of other $\Box_2$, with their surrogate variables $q_{\Box_2\psi}$. So $\varphi^{\mathbf{1}}$ is a unimodal formula containing only $\Box_1$. Let

$$\Theta^{\mathbf{1}}(\varphi) = \{p \mid p \text{ is a variable in } \varphi\} \cup \{\chi \in sub\,\Box_2\psi \mid \Box_2\psi \in sub\,\varphi\}.$$

The formula $\varphi^{\mathbf{2}}$ and the set $\Theta^{\mathbf{2}}(\varphi)$ are defined symmetrically.

Suppose now that φ is satisfiable in a model based on a frame for L. We need to construct a frame in $\mathcal{C}_1 \otimes \mathcal{C}_2$ satisfying φ. As we know only how to build frames for the unimodal fragments of L, the frame is constructed step-by-step alternating between $\Box_1$ and $\Box_2$.

Note first that since L_1 is characterised by $\mathcal{C}_1$, there is a model $\mathfrak{M}$ based on a frame in $\mathcal{C}_1$ and satisfying $\varphi^{\mathbf{1}}$ at a point r. Our aim now is to ensure that the formulas of the form $\Box_2\psi$ have the same truth-values as their surrogates $q_{\Box_2\psi}$. To do this, with each point x in $\mathfrak{M}$ we can associate the formula

$$\varphi_x = \bigwedge\{\psi \in \Theta^{\mathbf{1}}(\varphi) \mid (\mathfrak{M}, x) \models \psi^{\mathbf{1}}\} \wedge \bigwedge\{\neg\psi \mid \psi \in \Theta^{\mathbf{1}}(\varphi),\ (\mathfrak{M}, x) \not\models \psi^{\mathbf{1}}\},$$

construct a model $\mathfrak{M}_x$ based on a frame in $\mathcal{C}_2$ and satisfying $\varphi_x^{\mathbf{2}}$ in a world y, and then hook $\mathfrak{M}_x$ to $\mathfrak{M}$ by identifying x and y. After that we can switch to $\Box_1$ and in the same manner ensure that formulas $\Box_1\psi$ have the same truth-values as $q_{\Box_1\psi}$ at all points in every $\mathfrak{M}_x$, and so on. In this construction we use the fact that $\mathcal{C}_1$ and $\mathcal{C}_2$ are closed under isomorphic copies and disjoint unions: the $\mathfrak{M}_x$ should be mutually disjoint and the final model is the union of the models constructed at each step. Note that this construction is a special case of *fibring semantics* that is called iterated *dovetailing* [19, 20].

However, to realise this quite obvious scheme, we must be sure that $\varphi_x^{\mathbf{2}}$ is really satisfiable in a frame for L_2, which may impose some restrictions on the models we choose. First, in the construction above it is enough to deal with points x accessible from r in at most $md(\varphi)$ steps; no other point has any influence on the truth of φ at r. Let X be the set of all such points. Now, a sufficient and necessary condition for φ_x to be satisfiable in a frame for L (and so for $\varphi_x^{\mathbf{2}}$ to be satisfiable in a frame for L_2) can be formulated using the following general description of formulas of type φ_x.

Suppose Γ is a finite set of formulas closed under subformulas. Define the *consistency-set* $C(\Gamma)$ *of* Γ by taking

$$C(\Gamma) = \{\psi_\Delta \mid \Delta \subseteq \Gamma\},$$

where for $\Delta \subseteq \Gamma$,

$$\psi_\Delta = \bigwedge\{\chi \mid \chi \in \Delta\} \wedge \bigwedge\{\neg\chi \mid \chi \in \Gamma - \Delta\}.$$

In particular, for all $x \in X$, we have $\varphi_x \in C(\Theta^{\mathbf{1}}(\varphi))$. Given a formula φ, define

$$\Sigma_1(\varphi) = \{\psi \in C(\Theta^{\mathbf{1}}(\varphi)) \mid \neg\psi \notin L\}, \qquad \Sigma_2(\varphi) = \{\psi \in C(\Theta^{\mathbf{2}}(\varphi)) \mid \neg\psi \notin L\}.$$

The formulas in $\Sigma_i(\varphi)$ can be regarded as 'state descriptions' of the points in the possible models with respect to the formulas in $\Theta^{\mathbf{i}}(\varphi)$. In particular, for all $x \in X$, φ_x is satisfiable in a frame for L iff $\varphi_x \in \Sigma_1(\varphi)$. In other words, we should start with a model $\mathfrak{M}$ satisfying $\varphi^{\mathbf{1}} \wedge \Box_1^{\leq md(\varphi)}(\bigvee \Sigma_1(\varphi))^{\mathbf{1}}$ at a point r. Of course, the subsequent models $\mathfrak{M}_x$ must satisfy $\varphi_x^{\mathbf{2}} \wedge \Box_2^{\leq md(\varphi)}(\bigvee \Sigma_2(\varphi_x))^{\mathbf{2}}$ at all points $x \in X$, and so on. The interested reader may find more details in [48], [16].

Since the closure under *finite* disjoint unions is enough when we work with finite frames, we obtain the following:

THEOREM 4. *If both L_1 and L_2 are modal logics having the finite model property, then their fusion $L_1 \otimes L_2$ has the finite model property as well.*

As is shown by Wolter [82], *decidability* of the components also transfers to their fusion:

THEOREM 5. *If L_1 and L_2 are both decidable modal logics then $L_1 \otimes L_2$ is decidable as well.*

Further results showing that other important properties (such as Halldén completeness, decidability of the global consequence relation, uniform interpolation property) of modal logics are preserved under fusions were obtained in [48, 82].

As is discussed in Chapter 6 of this handbook, from the algebraic point of view every modal logic L can be regarded as the equational theory of modal algebras generated by the equations $\{`\varphi = 1' \mid \varphi \in L\}$. Thus, the problem of whether decidability is preserved under the formation of fusions of modal logics is an instance of the more general question: under which conditions does the decidability of two equational theories T_1 and T_2 imply the decidability of the union $T_1 \cup T_2$. The shared Boolean connectives impose special conditions on these equational theories; see the results of Ghilardi [29] that put the fusion construction to this more general context. Other extensions of Theorem 5 to fusions sharing not only the Booleans but also a universal modality and nominals are discussed in [30], and to fusions of non-normal modal logics in [6, 4].

2.2 Complexity of fusions

Unlike the properties considered above, upper complexity bounds do not always transfer under the formation of fusions (the lower bounds are inherited by Theorem 2 as long as we take fusions of consistent logics). The known decision procedures provide a time complexity bound for the fusion that is non-deterministic and one exponent higher than the maximal time complexity of the components. However, in general it is not known whether this increase in complexity is unavoidable. In particular, it is not known whether PSPACE- or EXPTIME-completeness transfers under the formation of fusions (see Theorem 7 below for some special cases when it actually does).

The following characterisation of the transfer of coNP-completeness was given by Spaan [77]. In order to formulate her theorem, we require the following notion. Say that a frame $\langle W', R' \rangle$ is a *skeleton subframe* of a frame $\langle W, R \rangle$ if $W' \subseteq W$ and $R' \subseteq R$. We use $\circ$ to denote reflexive points and $\bullet$ for irreflexive ones.

THEOREM 6. *Suppose that the unimodal logics L_1 and L_2 are characterised by classes $\mathcal{C}_1$ and $\mathcal{C}_2$ of frames, respectively, that are closed under the formation of isomorphic copies and disjoint unions. Then there are the following three cases for the complexity of $L_1 \otimes L_2$ (below $\{i, j\} = \{1, 2\}$):*

(1) $L_1 \otimes L_2$ *is* CONP-*complete.*

(2) $\mathcal{C}_i$ *consists of disjoint unions of singleton frames. In this case* $L_1 \otimes L_2$ *is polynomially reducible to* $\mathsf{Log}(\mathcal{C}_j)$.

(3) $L_1 \otimes L_2$ *is* PSPACE-*hard, whenever one of the following six cases holds:*

(i) $\bullet\!\leftarrow\!\bullet\!\rightarrow\!\bullet$ *and* $\bullet\!\rightarrow\!\bullet$ *are skeleton subframes of some frames in* $\mathcal{C}_i$ *and* $\mathcal{C}_j$, *respectively;*

(ii) $\circ\!\rightarrow\!\bullet\!\rightarrow\!\bullet$ *and* $\bullet\!\rightarrow\!\bullet$ *are skeleton subframes of some frames in* $\mathcal{C}_i$ *and* $\mathcal{C}_j$, *respectively;*

(iii) $\bullet\!\rightarrow\!\circ\!\rightarrow\!\bullet$ *and* $\bullet\!\rightarrow\!\bullet$ *are skeleton subframes of some frames in* $\mathcal{C}_i$ *and* $\mathcal{C}_j$, *respectively;*

(iv) $\bullet\!\rightarrow\!\bullet\!\rightarrow\!\bullet$ *and* $\circ\!\rightarrow\!\bullet$ *are skeleton subframes of some frames in* $\mathcal{C}_i$ *and* $\mathcal{C}_j$, *respectively;*

(v) $\bullet\!\rightarrow\!\bullet\!\rightarrow\!\bullet$ *and* $\circ\!\rightarrow\!\bullet$ *are skeleton subframes of a frame in* $\mathcal{C}_i$ *and* $\bullet\!\leftrightarrow\!\bullet$ *is a skeleton subframe of a frame in* $\mathcal{C}_j$;

(vi) $\bullet\!\rightarrow\!\bullet\!\rightarrow\!\bullet$ *and* $\circ\!\rightarrow\!\bullet$ *are skeleton subframes of a frame in* $\mathcal{C}_i$ *and* $\bullet\!\rightarrow\!\circ$ *is a skeleton subframe of a frame in* $\mathcal{C}_j$.

A close inspection of this result shows that almost all interesting fusions are PSPACE-hard. (An exception is the fusion **Alt** $\otimes$ **Alt** of two **Alt** logics that is CONP-complete by Theorem 6. We remind the reader that **Alt** is the CONP-complete logic determined by all *functional* frames.) In fact, the proof of Halpern and Moses [35] can be easily modified to obtain the following result on a matching upper bound for several 'standard' fusions:

THEOREM 7. *Let* $n > 1$ *and* $L_i \in \{\mathbf{K}, \mathbf{T}, \mathbf{K4}, \mathbf{S4}, \mathbf{KD45}, \mathbf{S5}\}$, *for all* $1 \leq i \leq n$. *Then* $L_1 \otimes \cdots \otimes L_n$ *is* PSPACE-*complete.*

Note that while **K**, **T**, **K4** and **S4** are PSPACE-complete themselves, **KD45** and **S5** are CONP-complete.

3 PRODUCT OF MODAL LOGICS

The formation of Cartesian products of various structures—vector and topological spaces, algebras, etc.—is a standard mathematical way of capturing the multidimensional character of our world. In modal logic, products of Kripke frames are natural constructions allowing us to reflect interactions between modal operators representing time, space, knowledge, actions, etc. The product construction as a combination method on modal logics was introduced in [74, 75, 24] and has been used in applications in computer science and artificial intelligence ever since (see, e.g., [68, 15, 7, 69, 18], and [23] and references therein).

DEFINITION 8. The *product* of two n-frames frames $\mathfrak{F}_1 = \langle W_1, R_1^1, \dots, R_1^n \rangle$ and $\mathfrak{F}_2 = \langle W_2, R_2^1, \dots, R_2^m \rangle$ is the $(n+m)$-frame

$$\mathfrak{F}_1 \times \mathfrak{F}_2 = \langle W_1 \times W_2, R_h^1, \dots, R_h^n, R_v^1, \dots, R_v^m \rangle$$

where $W_1 \times W_2 = \{\langle u, v\rangle \mid u \in W_1,\ v \in W_2\}$ and, for all $u_1, u_2 \in W_1$ and $v_1, v_2 \in W_2$,

$$\begin{array}{lll} \langle u_1, v_1\rangle\, R_h^i\, \langle u_2, v_2\rangle & \text{iff} & u_1 R_1^i u_2 \ \text{ and } \ v_1 = v_2 \quad (1 \leq i \leq n), \\ \langle u_1, v_1\rangle\, R_v^j\, \langle u_2, v_2\rangle & \text{iff} & u_1 = u_2 \ \text{ and } \ v_1 R_2^j v_2 \quad (1 \leq j \leq m). \end{array}$$

Such a frame will be called a *product frame*. The subscripts h and v appeal to the geometrical intuition of considering the R_h^i as 'horizontal' accessibility relations in $\mathfrak{F}_1 \times \mathfrak{F}_2$ and the R_v^j as 'vertical' ones; see Fig. 1 for an illustration.

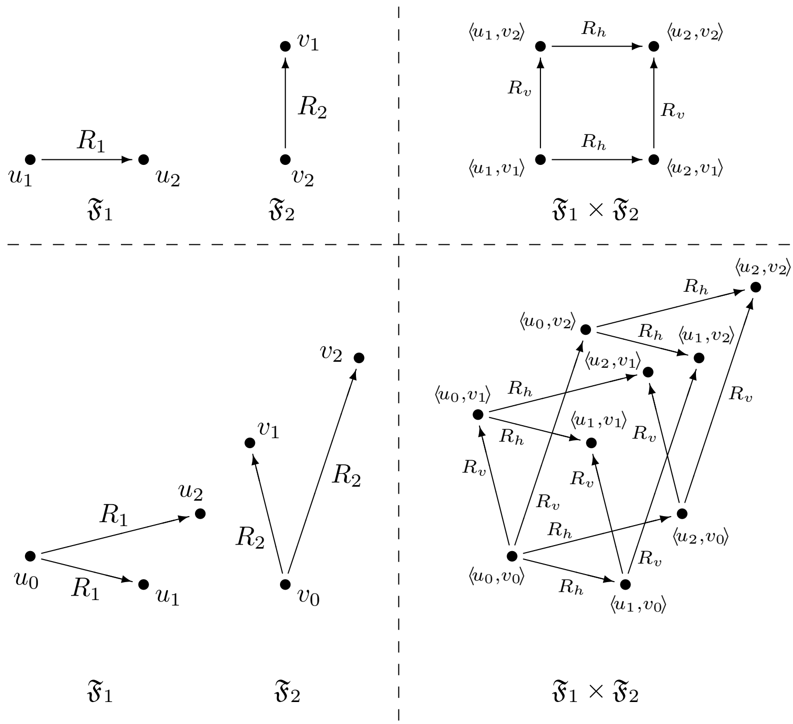


Figure 1. Product frames.

It is not hard to see that the product construction commutes with the three basic operations on frames:

PROPOSITION 9. *For all frames $\mathfrak{F}$, $\mathfrak{G}$, $\mathfrak{H}$, $\mathfrak{H}_i$, $i \in I$, the following hold:*

(i) *If $\mathfrak{F}$ is a p-morphic image of $\mathfrak{H}$, then $\mathfrak{F} \times \mathfrak{G}$ is a p-morphic image of $\mathfrak{H} \times \mathfrak{G}$.*

(ii) *If $\mathfrak{F}$ is a generated subframe of $\mathfrak{H}$, then $\mathfrak{F} \times \mathfrak{G}$ is a generated subframe of $\mathfrak{H} \times \mathfrak{G}$.*

(iii) *If $\mathfrak{F}$ is a disjoint union of $\mathfrak{H}_i$, $i \in I$, then $\mathfrak{F} \times \mathfrak{G}$ is isomorphic to the disjoint union of $\mathfrak{H}_i \times \mathfrak{G}$, $i \in I$.*

Products of Kripke frames can be used to define a natural combination method on modal logics:

DEFINITION 10. Let L_1 and L_2 be two Kripke complete modal logics formulated in languages $\mathcal{ML}_n$ and $\mathcal{ML}_m$ in such a way that they have disjoint sets of modal operators (say, $\Box_1, \ldots, \Box_n$ and $\Box_{n+1}, \ldots, \Box_{n+m}$, respectively). Then the *product of L_1 and L_2* is the modal logic

$$L_1 \times L_2 \;=\; \mathsf{Log}\{\mathfrak{F}_1 \times \mathfrak{F}_2 \mid \mathfrak{F}_i \in \mathsf{Fr}L_i,\ i = 1, 2\}.$$

For example, $\mathbf{K}_n \times \mathbf{K}_m$ is the $(n+m)$-modal logic determined by all product frames $\mathfrak{F}_1 \times \mathfrak{F}_2$, where $\mathfrak{F}_1$ is an n-frame and $\mathfrak{F}_2$ an m-frame; $\mathbf{S4} \times \mathbf{S5}$ is the bimodal logic determined by all product frames $\mathfrak{F}_1 \times \mathfrak{F}_2$ such that $\mathfrak{F}_1$ is reflexive and transitive, and $\mathfrak{F}_2$ is an equivalence frame.

Note that the product of Kripke complete modal logics is always Kripke complete by definition. It is important to emphasise that in order to make the product construction a well-defined combination method on Kripke complete modal logics, we have to consider products of *all* possible Kripke frames for L_1 and L_2. The reason is that even if $\mathsf{Log}\,\mathcal{C}_1 = \mathsf{Log}\,\mathcal{C}_1'$ and $\mathsf{Log}\,\mathcal{C}_2 = \mathsf{Log}\,\mathcal{C}_2'$, then we can have

$$\mathsf{Log}\{\mathfrak{F}_1 \times \mathfrak{F}_2 \mid \mathfrak{F}_i \in \mathcal{C}_i,\ i=1,2\} \ \neq\ \mathsf{Log}\{\mathfrak{F}_1 \times \mathfrak{F}_2 \mid \mathfrak{F}_i \in \mathcal{C}_i',\ i=1,2\},$$

see [23] for examples.

There are several attempts for extending the product construction from Kripke complete logics to arbitrary modal logics, mainly by considering product-like constructions on Kripke models, see [37, 23]. All the suggested methods so far result in sets of formulas that are not closed under the rule of Substitution, thus do not satisfy our criterion **(C2)**. Van Benthem *et al.* [79] show that by defining a product-like operator on their topological semantics, one can get back the *fusion* of modal logics determined by transitive frames.

Once the two-dimensional definition is given, there are essentially two ways of defining products of three or more modal logics. First, we can generalise in a straightforward way the definitions above. To simplify notation, from now on we will mostly consider products of unimodal frames and logics only. (However, we will discuss the multimodal versions in those cases when it does make a difference.)

DEFINITION 11. Given a natural number $n > 1$, the *product* of frames $\mathfrak{F}_1 = \langle W_1, R_1\rangle$, $\mathfrak{F}_2 = \langle W_2, R_2\rangle$, ..., $\mathfrak{F}_n = \langle W_n, R_n\rangle$ is the n-frame

$$\mathfrak{F}_1 \times \cdots \times \mathfrak{F}_n \ =\ \big\langle W_1 \times \cdots \times W_n, \bar{R}_1, \ldots, \bar{R}_n\big\rangle$$

where, for each $i = 1, \ldots, n$, $\bar{R}_i$ is a binary relation on $W_1 \times \cdots \times W_n$ such that

$$\langle u_1, \ldots, u_n\rangle\, \bar{R}_i\, \langle v_1, \ldots, v_n\rangle \quad \text{iff} \quad u_i R_i v_i \ \text{ and } \ u_k = v_k, \quad \text{for } k \neq i.$$

Then, given Kripke complete (uni)modal logics L_i formulated in the language having $\Box_i$ $(i = 1, \ldots, n)$, the *product of* $L_1, \ldots, L_n$ is the n-modal logic

$$L_1 \times \cdots \times L_n \ =\ \mathsf{Log}\{\mathfrak{F}_1 \times \cdots \times \mathfrak{F}_n \mid \mathfrak{F}_i \in \mathsf{Fr}L_i,\ i = 1, \ldots, n\}.$$

For example, $\mathbf{K}^n = \overbrace{\mathbf{K} \times \cdots \times \mathbf{K}}^{n}$ is the logic determined by all n-dimensional product frames; $\mathbf{S5}^n$ is the logic determined by all product frames $\mathfrak{F}_1 \times \cdots \times \mathfrak{F}_n$, where each $\mathfrak{F}_i$ is a (possible different) equivalence frame.

The second way would be to define $L_1 \times \cdots \times L_n$ as $(((L_1 \times L_2) \times L_3) \times \cdots \times L_{n-1}) \times L_n$. The easily established fact that the frame $\mathfrak{F}_1 \times \cdots \times \mathfrak{F}_n$ is isomorphic to

$$(((\mathfrak{F}_1 \times \mathfrak{F}_2) \times \mathfrak{F}_3) \times \cdots \times \mathfrak{F}_{n-1}) \times \mathfrak{F}_n$$

might seem to suggest that the two definitions are equivalent. However, the situation is not that simple. For example, it is not known whether the equalities

$$\mathbf{K}^4 \ \stackrel{?}{=}\ \mathbf{K}^3 \times \mathbf{K} \qquad \text{and} \qquad \mathbf{S5}^4 \ \stackrel{?}{=}\ \mathbf{S5}^3 \times \mathbf{S5}$$

hold. The problem here is that $\mathbf{K}^4$ is characterised by the class of products of four 1-frames, while $\mathbf{K}^3 \times \mathbf{K}$ by the class of products of *arbitrary* (that is, not necessarily product) 3-frames for $\mathbf{K}^3$ and 1-frames for $\mathbf{K}$. Now, the thing is that these arbitrary $\mathbf{K}^3$-frames are not necessarily isomorphic to product frames (in fact, we do not even know what they look like; see Theorem 25).

For this reason, we take Definition 11 above as the 'official' definition of higher dimensional product logics. Note, however, that in Section 3.3 we provide a characterisation of arbitrary (countable) frames for $\mathbf{K}\times\mathbf{K}$ and $\mathbf{S5}\times\mathbf{S5}$ (among many other two-dimensional product logics), and prove—with the help of this characterisation—that for many three-dimensional products the two definitions actually coincide. For instance,

$$\mathbf{K}^3 \;=\; (\mathbf{K}\times\mathbf{K})\times\mathbf{K} \qquad \text{and} \qquad \mathbf{S5}^3 \;=\; (\mathbf{S5}\times\mathbf{S5})\times\mathbf{S5},$$

see Corollary 23.

3.1 General transfer results

Compared to fusions, there are very few general transfer results for products. In fact, as we shall see in Sections 3.3 and 3.4, for many cases the lack of transfer of finite axiomatisability and decidability is the 'norm'.

In this section we discuss some basic properties of the product construction and the very few general transfer results about it. To begin with, observe that in the definition of product logics it is enough to consider only *rooted* frames for the component logics. Indeed, the inclusion

$$L_1\times\cdots\times L_n \;\subseteq\; \mathsf{Log}\{\mathfrak{F}_1\times\cdots\times\mathfrak{F}_n \mid \mathfrak{F}_i \text{ is a rooted frame for } L_i,\ i=1,\dots,n\}$$

should be clear. To show the converse, suppose $\varphi\notin L_1\times\cdots\times L_n$, i.e., φ is refuted at a point $\langle u_1,\dots,u_n\rangle$ in some model based on a product frame $\mathfrak{F}_1\times\cdots\times\mathfrak{F}_n$, where $\mathfrak{F}_i$ is a frame for L_i, $i=1,\dots,n$. For each i, let $\mathfrak{G}_i$ be the subframe of $\mathfrak{F}_i$ generated by u_i. Then $\mathfrak{G}_i$ is also a frame for L_i, for $i=1,\dots,n$. On the other hand, it is readily checked that $\mathfrak{G}_1\times\cdots\times\mathfrak{G}_n$ is isomorphic to the subframe of $\mathfrak{F}_1\times\cdots\times\mathfrak{F}_n$ generated by $\langle u_1,\dots,u_n\rangle$. Thus we obtain the following:

PROPOSITION 12. *For all Kripke complete modal logics $L_1,\dots,L_n$,*

$$L_1\times\cdots\times L_n \;=\; \mathsf{Log}\{\mathfrak{F}_1\times\cdots\times\mathfrak{F}_n \mid \mathfrak{F}_i \textit{ is a rooted frame for } L_i,\ i=1,\dots,n\}.$$

For instance, $\mathbf{S5}^n$ is determined by products of universal frames $\langle W_i, W_i\times W_i\rangle$, $i=1,\dots,n$. Moreover, each such 'universal product frame' is a p-morphic image of a *cubic universal product frame*, i.e., the nth power of the same universal frame $\langle W, W\times W\rangle$. Indeed, it is easy to see that if a set W is such that there are surjections $f_i : W\to W_i$, for $i=1,\dots,n$, then the map f defined by

$$f(w_1,\dots,w_n)=\langle f_1(w_1),\dots,f_n(w_n)\rangle$$

is a p-morphism from the frame $\langle W, W\times W\rangle^n$ onto

$$\langle W_1, W_1\times W_1\rangle\times\cdots\times\langle W_n, W_n\times W_n\rangle\,.$$

Such a set and surjections can be found, for example, by taking the disjoint union of the W_i as W and defining f_i so that it is the identity map on W_i and arbitrary otherwise. Therefore, we obtain:

PROPOSITION 13. $\mathbf{S5}^n$ *is determined by the class of all cubic universal product frames.*

The formation of products as a combination method satisfies criterion **(C3)**, as the following proposition shows:

PROPOSITION 14. *For all Kripke complete modal logics $L_1, \dots, L_n$,*

$$L_1 \otimes \cdots \otimes L_n \ \subseteq \ L_1 \times \cdots \times L_n.$$

Proof. Given a product frame $\mathfrak{F}_1 \times \cdots \times \mathfrak{F}_n = \left\langle W_1 \times \cdots \times W_n, \bar{R}_1, \dots, \bar{R}_n \right\rangle$ such that each $\mathfrak{F}_i = \langle W_i, R_i \rangle$ is a frame for L_i $(i = 1, \dots, n)$, fix some $1 \leq i \leq n$. For every $n-1$-tuple $\bar{u}_i = \langle u_1, \dots, u_{i-1}, u_{i+1}, \dots, u_n \rangle$ with $u_j \in W_j$, for $j \neq i$, we take the set

$$W_{\bar{u}_i} = \{\langle u_1, \dots, u_n \rangle \mid u_i \in W_i,\ \langle u_1, \dots, u_{i-1}, u_{i+1}, \dots, u_n \rangle = \bar{u}_i\},$$

and let $S_{\bar{u}_i}$ be the restriction of $\bar{R}_i$ to $W_{\bar{u}_i}$, i.e., $S_{\bar{u}_i} = \bar{R}_i \cap (W_{\bar{u}_i} \times W_{\bar{u}_i})$. Then we have:

- $\langle W_{\bar{u}_i}, S_{\bar{u}_i} \rangle$ is isomorphic to $\langle W_i, R_i \rangle$;
- $\left\langle W_1 \times \cdots \times W_n, \bar{R}_i \right\rangle$ is the disjoint union of the frames $\langle W_{\bar{u}_i}, S_{\bar{u}_i} \rangle$, for all $n-1$-tuples $\bar{u}_i$.

❑

As we shall see in Section 3.3, the inclusion in Proposition 14 is proper: product logics always include certain *interactions* between the modal operators of their components. Note, however, that the modal operators within each component are not affected by these interactions, that is, the product $L_1 \times \cdots \times L_n$ of consistent Kripke complete logics $L_1, \dots, L_n$ is a *conservative extension* of each of them. One can even show a slightly stronger statement:

PROPOSITION 15. *Let $L_1, \dots, L_n, L_{n+1}$ be consistent Kripke complete unimodal logics. Then the logic $L_1 \times \cdots \times L_n \times L_{n+1}$ is a conservative extension of $L_1 \times \cdots \times L_n$, i.e., for every $\mathcal{ML}_n$-formula φ,*

$$\varphi \in L_1 \times \cdots \times L_n \quad \textit{iff} \quad \varphi \in L_1 \times \cdots \times L_n \times L_{n+1}.$$

Proof. We prove this only for the case $L_1 = \cdots = L_n = L$; the general case is considered in a similar way. First, it is readily checked that for any $n+1$-dimensional product frame

$$\mathfrak{F} = \left\langle W_1 \times \cdots \times W_n \times W_{n+1}, \bar{R}_1, \dots, \bar{R}_n, \bar{R}_{n+1} \right\rangle,$$

the projection map $f(w_1, \dots, w_n, w_{n+1}) = \langle w_1, \dots, w_n \rangle$ is a p-morphism from the 'n-reduct'

$$\mathfrak{F}_{(n)} = \left\langle W_1 \times \cdots \times W_n \times W_{n+1}, \bar{R}_1, \dots, \bar{R}_n \right\rangle$$

of $\mathfrak{F}$ onto the n-dimensional product frame $\mathfrak{F}^- = \left\langle W_1 \times \cdots \times W_n, \bar{R}_1, \dots, \bar{R}_n \right\rangle$.

Now suppose that $\varphi \in L^{n+1}$ and $\mathfrak{G}$ is an n-dimensional product frame for L^n. As L is consistent and Kripke complete, there exists a frame $\mathfrak{H}$ for L. Then the product $\mathfrak{F} = \mathfrak{G} \times \mathfrak{H}$ is a frame for L^{n+1}, and so $\mathfrak{F} \models \varphi$. Since $\mathfrak{F}^- = \mathfrak{G}$, we finally obtain $\mathfrak{G} \models \varphi$.

Conversely, suppose that $\varphi \in L^n$, and let $\mathfrak{F}$ be an $n+1$-dimensional product frame for L^{n+1}. Then clearly $\mathfrak{F}^-$ is a frame for L^n, and so $\mathfrak{F} \models \varphi$. ❑

A useful property of certain product logics that sometimes they are determined by their *countable* product frames:

THEOREM 16. *Let L_i be a Kripke complete unimodal logic such that $\mathsf{Fr}L_i$ is first-order definable in the language having equality and a binary predicate symbol R_i, for each $i = 1, \dots, n$. Then $L_1 \times \cdots \times L_n$ is determined by the class of its countable product frames.*

Proof. For each i, let Γ_i denote the first-order theory defining $\mathsf{Fr}L_i$ in the language $\mathcal{L}_n$ having equality and binary predicate symbols $R_1, \dots, R_n$. Now let $\mathcal{L}_n^\times$ be the $n+1$-sorted extension of $\mathcal{L}_n$ that has the binary predicate symbols $R_1, \dots, R_n$ of sort 0, countably many unary predicate symbols $P_0, P_1, \dots$ of sort 0, and for each sort i ($i = 1, \dots, n$) a unary function symbol f_i taking an argument of sort 0 and returning a value of sort i. For each $\phi \in \Gamma_i$, denote by ϕ' the formula obtained by substituting $f_i(x)$ for all occurrences of each variable x in ϕ ($i = 1, \dots, n$). Let

$$\Sigma = \{\phi' \mid \phi \in \Gamma_i,\ i = 1, \dots, n\} \cup \{\pi\},$$

where π is the following sentence:

$$\begin{aligned} &\forall x \forall y \ \big(f_1(x) = f_1(y) \wedge \cdots \wedge f_n(x) = f_n(y) \to x = y\big) \\ &\qquad \wedge \forall x_1 \dots \forall x_n \exists y \ \big(f_1(y) = x_1 \wedge \cdots \wedge f_n(y) = x_n\big) \\ &\qquad\qquad \wedge \bigwedge_{i=1}^{n} \forall x \forall y \ \Big(x R_i y \ \leftrightarrow \ \big(f_i(x) R_i f_i(y) \wedge \bigwedge_{\substack{j=1 \\ j \neq i}}^{n} f_j(x) = f_j(y)\big)\Big) \end{aligned}$$

(here x and y are variables of sort 0, and x_i is of sort i, for $i = 1, \dots, n$). Now suppose that $\varphi \notin L_1 \times \cdots \times L_n$, for some $\mathcal{ML}_n$-formula φ. Then φ is not true in a model $\mathfrak{M} = \langle \mathfrak{F}, \mathfrak{V} \rangle$ based on the product $\mathfrak{F}_1 \times \cdots \times \mathfrak{F}_n$ of frames $\mathfrak{F}_i = \langle W_i, S_i \rangle$ such that $\mathfrak{F}_i \models \Gamma_i$ for $i = 1, \dots, n$. Define a first-order $\mathcal{L}_n^\times$-structure I by taking

$$I = \big\langle W_1 \times \cdots \times W_n, W_1, \dots, W_n; \bar{S}_1, \dots, \bar{S}_n, \mathfrak{V}(p_0), \mathfrak{V}(p_1), \dots, pr_1, \dots, pr_n \big\rangle,$$

where $pr_i : W_1 \times \cdots \times W_n \to W_i$ are the projection functions. It is easy to see that $I \models \Sigma$. Since without the extra sorts and the projections I is nothing but the modal model $\mathfrak{M}$ considered as a first-order structure, we also have $I \not\models \forall x \varphi^\star(x)$ (where $\varphi^\star$ is the *standard translation* of φ). In other words, $\Sigma' = \Sigma \cup \{\exists x \neg \varphi^\star(x)\}$ is true in I. By the downward Löwenheim–Skolem–Tarski theorem, there is a countable first-order $\mathcal{L}_n^\times$-structure

$$J = \big\langle U, U_1, \dots, U_n; , R_1^J, \dots, R_n^J, P_0^J, P_1^J, \dots, f_1^J, \dots, f_n^J \big\rangle$$

such that $J \models \Sigma'$. For each $i = 1, \dots, n$, define

$$Q_i = \{\big\langle f_i^J(u), f_i^J(v) \big\rangle \mid \langle u, v \rangle \in R_i^J\},$$

and for each $j < \omega$,

$$P_j^{I'} = \{\big\langle f_1^J(w), \dots, f_n^J(w) \big\rangle \mid w \in P_j^J\}.$$

Since $J \models \pi$, the map $h(w) = \big\langle f_1^J(w), \dots, f_n^J(w) \big\rangle$ is an isomorphism between J and the first-order $\mathcal{L}_n^\times$-structure

$$I' = \Big\langle U_1 \times \cdots \times U_n, U_1, \dots, U_n; \bar{Q}_1, \dots, \bar{Q}_n, P_0^{I'}, P_1^{I'}, \dots, pr_1, \dots, pr_n \Big\rangle.$$

Thus, $I' \models \Sigma$ and $I' \not\models \forall x \varphi^\star(x)$. Let $\mathfrak{G}_i = \langle U_i, Q_i \rangle$, $i = 1, \dots, n$. Define a valuation $\mathfrak{W}$ in the (countable) product frame $\mathfrak{G} = \mathfrak{G}_1 \times \cdots \times \mathfrak{G}_n$ by taking $\mathfrak{W}(p_j) = P_j^{I'}$ for $j < \omega$. As without the extra sorts and the projections I' is just a (countable) modal model $\mathfrak{N} = \langle \mathfrak{G}, \mathfrak{W} \rangle$ considered as a first-order structure, this means that φ is not true in $\mathfrak{N}$.

Note that in fact we have also proved that

$$\varphi \in L_1 \times \cdots \times L_n \qquad \text{iff} \qquad \Sigma \models \forall x \varphi^\star(x), \tag{1}$$

for any $\mathcal{ML}_n$-formula φ. ❑

In many cases recursive enumerability of the components transfers to their product:

THEOREM 17. *Let L_i be a Kripke complete unimodal logic such that $\mathsf{Fr}L_i$ is definable by a recursive set of first-order sentences in the language having equality and a binary predicate symbol R_i, for each $i = 1, \dots, n$. Then the product logic $L_1 \times \cdots \times L_n$ is recursively enumerable.*

Proof. We use the notation of the proof of Theorem 16. Since now the sets Γ_i are recursive, Σ is recursive as well. And since the consequence relation of first-order logic is recursively enumerable, it follows from (1) that $L_1 \times \cdots \times L_n$ is recursively enumerable. ❑

3.2 Connections with other formalisms

The product construction shows up in various disguises, here we discuss three examples: first-order logics, 'interpreted systems' for temporal epistemic logics, and modal extensions of description logics.

First-order classical and modal logics

Let us fix a natural number $n > 0$ and consider the fragment of classical first-order logic that

- uses n individual variables $x_1, \dots, x_n$,
- contains neither equality, nor individual constants, nor function symbols, and
- whose atomic formulas are of the form $P(x_1, \dots, x_n)$, where P is an n-ary predicate symbol.

This fragment can be regarded as the '*n-variable substitution- and equality-free fragment*' of classical first-order logic. The following map $\cdot^\bullet$ provides a one-to-one correspondence between formulas of this fragment and $\mathcal{ML}_n$-formulas:

$$\begin{array}{rclcrcl} P_i(x_1, \dots, x_n)^\bullet & = & p_i & \qquad & (\varphi \wedge \psi)^\bullet & = & \varphi^\bullet \wedge \psi^\bullet, \\ (\neg \varphi)^\bullet & = & \neg \varphi^\bullet, & & (\exists x_i \psi)^\bullet & = & \Diamond_i \psi^\bullet \qquad (1 \le i \le n). \end{array}$$

It is not hard to see that, for every first-order formula of the fragment,

$$\varphi \text{ is first-order valid} \quad \text{iff} \quad \varphi^\bullet \in \mathbf{S5}^n.$$

Indeed, every first-order structure $I = \left\langle D^I, \dots, P_i^I, \dots \right\rangle$ can be considered as a modal model $\mathfrak{M}(I) = \langle \langle W, \dots, R_i, \dots \rangle, \mathfrak{V} \rangle$, where

- W is the set of all variable assignments in I, i.e., the set of all functions from the variables $x_1, \dots, x_n$ into D^I;
- $\mathfrak{a} R_i \mathfrak{b}$ iff $\mathfrak{a}(x_j) = \mathfrak{b}(x_j)$ for all variables x_j different from x_i, $1 \le i \le n$;
- $\mathfrak{V}(p_i) = P_i^I$.

The set W of all assignments in I can be regarded as the n^{th} Cartesian power of the domain D^I. The underlying frame of $\mathfrak{M}(I)$ then turns into a product frame for $\mathbf{S5}^n$: the nth power of the universal $\mathbf{S5}$-frame $\langle D^I, D^I \times D^I \rangle$. On the other hand, $\mathbf{S5}^n$ is determined by such cubic universal product frames by Proposition 13.

The idea of such a 'modal approach' to classical first-order logic was suggested by Quine [66] and Kuhn [51] and fully realised by Venema [80]. 'Approximating' first-order logic with logical systems of propositional character was an important motive in the *algebraic* treatment of classical first-order logic; see the work of Tarski and his school [38, 39, 1, 11, 13, 34, 62]. The modal algebras (see Chapter 6 of this handbook) corresponding to the product logic $\mathbf{S5}^n$ are known in the algebraic logic literature as *diagonal-free cylindric set algebras of dimension n.*

As is shown in [23], a similar connection can be established between n-variable fragments of quantified modal logics L (with constant domains) and $n+1$-dimensional product logics of the form

$$L \times \overbrace{\mathbf{S5} \times \cdots \times \mathbf{S5}}^{n}.$$

Temporal epistemic logics

Here we briefly discuss the connections to the 'interpreted systems' approach proposed by Fagin *et al*. [15] which gives rise to various combinations of propositional temporal and epistemic logics ranging from fusions to products of these logics.

Suppose S is a non-empty set (of 'states') and $\mathfrak{F} = \langle T, < \rangle$ is a strict linear order (the 'flow of time'). Suppose also that $\mathcal{R}$ is a non-empty set of functions from T to S (the available 'runs of events' over $\mathfrak{F}$), and let $R_1, \dots, R_n$ be binary relations on $T \times \mathcal{R}$. Then the tuple

$$\mathfrak{S} = \langle T, \mathcal{R}, <, R_1, \dots, R_n \rangle$$

is called a *interpreted system.* A *valuation* $\mathfrak{V}$ *in* $\mathfrak{S}$ is a function from the set of propositional variables into the set 2^S of all subsets of S. The pair $\mathfrak{M} = \langle \mathfrak{S}, \mathfrak{V} \rangle$ is called a *model based on* $\mathfrak{S}$.

We interpret the modal language $\mathcal{ML}_{n+1}$ at $\langle$timepoint,run$\rangle$ pairs in these models. $\Box_1$ represents the temporal operator 'always in the future', while $\Box_2, \dots, \Box_{n+1}$ represent the respective knowledge of n agents:

- $(\mathfrak{M}, \langle t, f \rangle) \models p$ iff $f(t) \in \mathfrak{V}(p)$,
- $(\mathfrak{M}, \langle t, f \rangle) \models \varphi \wedge \psi$ iff $(\mathfrak{M}, \langle t, f \rangle) \models \varphi$ and $\langle t, f \rangle \models \psi$,
- $(\mathfrak{M}, \langle t, f \rangle) \models \neg\varphi$ iff not $(\mathfrak{M}, \langle t, f \rangle) \models \varphi$,
- $(\mathfrak{M}, \langle t, f \rangle) \models \Box_1\varphi$ iff $(\mathfrak{M}, \langle t', f \rangle) \models \varphi$ whenever $f' > f$,
- $(\mathfrak{M}, \langle t, f \rangle) \models \Box_i\varphi$ iff $(\mathfrak{M}, \langle t', f' \rangle) \models \varphi$ whenever $\langle t, f \rangle\, R_{i-1}\, \langle t', f' \rangle$ $(i = 2, \dots, n+1)$.

We say that φ *is true in* $\mathfrak{M}$ if $(\mathfrak{M}, \langle t, f\rangle) \models \varphi$ holds, for every $\langle t, f\rangle \in T \times \mathcal{R}$.

Given a propositional temporal logic $\mathsf{Log}\,\mathcal{C}_1$ determined by a class $\mathcal{C}_1$ of strict linear orders and an n-modal epistemic logic L determined by a class $\mathcal{C}_2$ of n-frames, we can obtain a 'combined' temporal-epistemic logic by considering all $\mathcal{ML}_{n+1}$-formulas that are true in all models that are based on interpreted systems of the form $\langle T, \mathcal{R}, <, R_1, \dots, R_n\rangle$ such that $\langle T, <\rangle \in \mathcal{C}_1$ and $\langle T \times \mathcal{R}, R_1, \dots, R_n\rangle \in \mathcal{C}_2$. By Theorem 3, this combined logic is just the *fusion* of $\mathsf{Log}\,\mathcal{C}_1$ and L.

By imposing various constraints on interpreted systems, we can reflect some interesting features of agents. An interpreted system $\mathfrak{S}$ models agents who *know the time* if, for all $t, t' \in T$, $f, f' \in \mathcal{R}$, and $i = 1, \dots, n$,

$$\langle t, f\rangle\, R_i\, \langle t', f'\rangle \text{ implies } t = t'.$$

In other words, if A_i believes that at moment t relative to an evolution f the pair $\langle t', f'\rangle$ represents a possible state of affairs, then $t = t'$. So at each moment t the agents are assumed to know that the clock is at t. Systems represented by structures of this type are known as *synchronous.*

An interpreted system models agents who *do not learn* if, for all agents A_i, $f, f' \in \mathcal{R}$ and $t, t' \in T$, we have

$$\langle t, f\rangle\, R_i\, \langle t', f'\rangle \text{ implies } \forall s \geq t\ \exists s' \geq t'\ \ \langle s, f\rangle\, R_i\, \langle s', f'\rangle\,.$$

Intuitively, an agent A_i does not learn if, whenever it regards w as a possible state of affairs at moment t, then it regards w as a possible state of affairs at every moment $s \geq t$ as well. Under the condition that agents know the time, this means that if agent A_i regards an evolution f' as possible at t then it regards f' as possible at every $s > t$. Similarly, an interpreted system models agents who *do not forget* if, for all A_i, $t, t' \in T$ and $f, f' \in \mathcal{R}$, we have

$$\langle t, f\rangle\, R_i\, \langle t', f'\rangle \text{ implies } \forall s \leq t\ \exists s' \leq t'\ \ \langle s, f\rangle\, R_i\, \langle s', f'\rangle\,.$$

Systems of this type are known also as systems with *perfect recall.*

If an interpreted system models agents who know time, do not forget and do not learn, then, for all agents A_i, $t, t' \in T$ and $f, f' \in \mathcal{R}$, we have

$$\langle t, f\rangle\, R_i\, \langle t', f'\rangle \text{ implies } t = t' \text{ and } \forall s\ \ \langle s, f\rangle\, R_i\, \langle s, f'\rangle\,.$$

Thus, the interpretation of $\mathcal{ML}_{n+1}$-formulas in $\mathfrak{S}$ corresponds to evaluating them in the *product* of frames $\mathfrak{F} = \langle T, <\rangle$ and $\langle \mathcal{R}, S_1, \dots, S_n\rangle$, where

$$f S_i f' \quad \text{iff} \quad \exists t, t' \in T\ \ \langle t, f\rangle\, R_i\, \langle t', f'\rangle \quad \text{iff} \quad \forall t \in T\ \ \langle t, f\rangle\, R_i\, \langle t, f'\rangle\,.$$

'Modal' description logics

As is discussed in Chapter 13 of this handbook, originally *description logics* have been designed and used as a formalism for knowledge representation and reasoning only in 'static' application domains. Later on, several attempts have been made in the literature in order to extend description logics with 'dynamic' features such as knowledge as time- or action-dependence, beliefs of different agents, etc. (see, e.g., [72, 71, 57, 32, 7, 3, 5,

84, 86, 88]). Here we briefly describe a simple 'modal' extension of the basic concept language $\mathcal{ALC}$ (see Chapter 13 of this handbook) and its connection to products.

Imagine, for instance, a car salesman John who, besides standard ABox and TBox knowledge bases (see Chapter 13 of this handbook), also wants to include 'modalised' concepts such as describing a $\mathsf{Customer}$ as

$$\mathsf{Homo\_sapiens} \sqcap \langle \text{sometime in the past} \rangle\, \exists \mathsf{buys.Car},$$

or a $\mathsf{Potential\_customer}$ as

$$[\text{John believes}]\, \langle \text{eventually} \rangle\, \mathsf{Customer}.$$

Concept descriptions in the extended concept language $\mathcal{ML}_n^{\mathcal{ALC}}$ that is able to express these can be formed according to the following rules:

$$C, D \quad \to \quad A \mid \top \mid \bot \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall r.C \mid \exists r.C \mid \Box_i C \mid \Diamond_i C,$$

where A ranges over concept names, r ranges over role names, and $i = 1, \dots, n$.

The intended semantics of $\mathcal{ML}_n^{\mathcal{ALC}}$ is defined as follows. An $\mathcal{ML}_n^{\mathcal{ALC}}$-*interpretation with constant domains and roles* is a pair $\mathfrak{M} = \langle \mathfrak{F}, I \rangle$ in which $\mathfrak{F} = \langle W, R_1, \dots, R_n \rangle$ is an n-frame and I is a function associating with each $w \in W$ a usual $\mathcal{ALC}$-interpretation

$$I(w) = \big\langle \Delta, \dots, A^{\mathfrak{M},w}, \dots, r, \dots \big\rangle$$

(that is, Δ is a nonempty set, $A^{\mathfrak{M},w} \subseteq \Delta$ for each concept name A, and $r \subseteq \Delta \times \Delta$ for each role name r). The (world-dependent) interpretation of concept names is inductively extended to arbitrary concept descriptions. Here we give the definition for the new 'modal' constructors only:

$$(\Box_i C)^{\mathfrak{M},w} = \bigcap_{wR_iv} C^{\mathfrak{M},v}, \qquad (\Diamond_i C)^{\mathfrak{M},w} = \bigcup_{wR_iv} C^{\mathfrak{M},v}.$$

Now given a Kripke complete n-modal logic L, we say that a concept description C is $L_{\mathcal{ALC}}$-*satisfiable* (with an empty knowledge base) if there is an $\mathcal{ML}_n^{\mathcal{ALC}}$-interpretation (with constant domains and roles) $\mathfrak{M} = \langle \mathfrak{F}, I \rangle$ and a world w in $\mathfrak{F}$ such that $\mathfrak{F}$ is a frame for L and $C^{\mathfrak{M},w} \neq \emptyset$.

Now, by extending the correspondence between $\mathcal{ALC}$ (with m role names) and the modal logic $\mathbf{K}_m$ (see Chapter 13 of this handbook), it is straightforward to see that $L_{\mathcal{ALC}}$-satisfiability coincides with $L \times \mathbf{K}_m$-satisfiability.

3.3 Axiomatising products

Product logics are defined in a semantical way: they are logics determined by classes of product frames, and so Kripke complete by definition. Therefore, the proper 'transfer' question to ask is how a possible axiomatisation for a product logic relates to axiomatisations of its components.

To begin with, observe that the following properties hold in every product frame of the form $\langle W, R_1, \dots, R_n \rangle$, for all $i, j = 1, \dots, n$, $i \neq j$:

- *left commutativity*: $\forall x, y, z \in W \big(xR_jy \wedge yR_iz \to \exists u \in W\,(xR_iu \wedge uR_jz)\big)$,

- *right commutativity*: $\forall x, y, z \in W \big(xR_iy \wedge yR_jz \to \exists u \in W\,(xR_ju \wedge uR_iz)\big)$,
- *Church–Rosser property*: $\forall x, y, z \in W \big(xR_jy \wedge xR_iz \to \exists u \in W\,(yR_iu \wedge zR_ju)\big)$,

see Fig. 2.

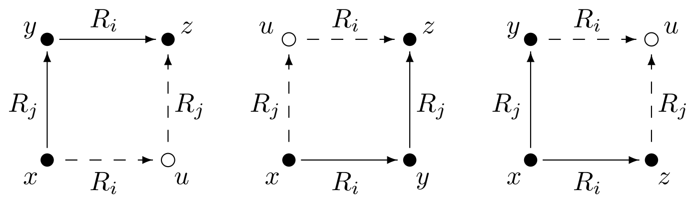


Figure 2. Left and right commutativity and Church–Rosser properties.

These properties can also be expressed by modal formulas. One can easily check that an arbirtary (not necessarily product) n-frame is left commutative iff it validates the formulas

$$\boldsymbol{com}^{\boldsymbol{l}}_{ij} \;=\; \Diamond_j\Diamond_i p \to \Diamond_i\Diamond_j p,$$

it is right commutative iff it validates

$$\boldsymbol{com}^{\boldsymbol{r}}_{ij} \;=\; \Diamond_i\Diamond_j p \to \Diamond_j\Diamond_i p,$$

and it is Church–Rosser iff it validates

$$\boldsymbol{chr}_{ij} \;=\; \Diamond_i\Box_j p \to \Box_j\Diamond_i p.$$

The corresponding left and right commutativity axioms can be combined into a single *commutativity* axiom

$$\boldsymbol{com}_{ij} \;=\; \boldsymbol{com}^{\boldsymbol{l}}_{ij} \wedge \boldsymbol{com}^{\boldsymbol{r}}_{ij}.$$

DEFINITION 18. Given modal logics L_i formulated in the language having $\Box_i$ ($i = 1, \dots, n$), the *commutator*

$$[L_1, \dots, L_n]$$

of $L_1, \dots, L_n$ is the smallest n-modal logic containing all the L_i and the axioms $\boldsymbol{com}_{ij}$ and $\boldsymbol{chr}_{ij}$, for all $i, j = 1, \dots, n$, $i \neq j$.

Note that the commutator of (finitely) axiomatisable modal logics is always (finitely) axiomatisable by definition. Moreover, since the axioms $\boldsymbol{com}_{ij}$ and $\boldsymbol{chr}_{ij}$ are Sahlqvist formulas, we also have:

PROPOSITION 19. *The commutator of canonical logics is canonical, and so Kripke complete.*

It is worth noting that even if all components are Kripke complete, their commutator is not necessarily so: non-examples are [**K4**, **GL.3**] and [**GL**, **Grz.3**], see Section 3.4.

Commutators are natural candidates for axiomatising products. As $\boldsymbol{com}_{ij}$ and $\boldsymbol{chr}_{ij}$ are valid in every product frame, by Proposition 14 we always have that

$$[L_1, \dots, L_n] \;\subseteq\; L_1 \times \cdots \times L_n, \tag{2}$$

whenever $L_1, \dots, L_n$ are Kripke complete modal logics. Those tuples of logics $L_1, \dots, L_n$ for which the converse inclusion also holds are called *product-matching*.

Axiomatising two-dimensional product logics

We begin with a general result of Gabbay and Shehtman [24] stating that certain pairs of modal logics are always product-matching.

Consider the first-order language with equality and a binary predicate R. A formula ψ in this language is called *positive* if it is built up from atoms using only $\wedge$ and $\vee$. A sentence of the form

$$\forall x \forall y \forall \bar{z} \, \big(\psi(x, y, \bar{z}) \to R(x, y)\big)$$

is said to be a *universal Horn sentence* if $\psi(x, y, \bar{z})$ is a positive formula. We call an $\mathcal{ML}$-formula φ a *Horn formula*, if there is a universal Horn sentence φ_H such that, for all frames $\mathfrak{F}$,

$$\mathfrak{F} \models \varphi \qquad \text{iff} \qquad \mathfrak{F} \models \varphi_H.$$

An $\mathcal{ML}$-formula is called *variable free* if it contains no propositional variables, i.e., all its atomic subformulas are constants $\bot$ or $\top$.

DEFINITION 20. A modal logic is called *Horn axiomatisable* if it is axiomatisable by only Horn and variable-free formulas.

It is not hard to see that if L is a Kripke complete and Horn axiomatisable logic then $\mathsf{Fr}L$ is defined by the set

$$\Gamma_L = \{\varphi_H \mid \varphi \text{ is a Horn axiom of } L\} \ \cup \ \{\varphi^\star \mid \varphi \text{ is a variable-free axiom of } L\} \qquad (3)$$

of first-order formulas (here $\varphi^\star$ is the *standard translation* of φ). Examples of Kripke complete Horn axiomatisable logics are **K**, **D**, **K4**, **S4**, **KD45**, **T**, **S5**.

THEOREM 21. *Let L_1 and L_2 be Kripke complete and Horn axiomatisable modal logics. Then*

$$L_1 \times L_2 \ = \ [L_1, L_2].$$

Proof. The heart of the proof is the following lemma that can be proved by constructing the necessary p-morphism in a step-by-step manner, see [23, Lemmas 5.2 and 5.8].

LEMMA 22. *Let L_1 and L_2 be Kripke complete and Horn axiomatisable unimodal logics. Then every countable rooted 2-frame for $[L_1, L_2]$ is a p-morphic image of a product frame for $L_1 \times L_2$.*

Now, by Proposition 19, $[L_1, L_2]$ is determined by the class of commutative and Church–Rosser frames from $\mathsf{Fr}(L_1 \otimes L_2)$. By (3), this class is first-order definable in the language with equality and two binary predicate symbols. Let $\varphi \notin [L_1, L_2]$. Then, using the standard translation $\varphi^\star$ of φ and the downward Löwenheim–Skolem–Tarski theorem, it is not hard to see that we can have a countable rooted 2-frame $\mathfrak{F}$ for $[L_1, L_2]$ refuting φ. Now, using Lemma 22, we can find a product frame $\mathfrak{G}$ for $L_1 \times L_2$ having $\mathfrak{F}$ as its p-morphic image. By Proposition 9, it follows that $\mathfrak{G} \not\models \varphi$, and so $\varphi \notin L_1 \times L_2$. Therefore, $L_1 \times L_2 \subseteq [L_1, L_2]$. The converse inclusion has already been shown as (2). ❑

As a corollary of Theorem 21 we obtain that finite axiomatisability of two Kripke complete and Horn axiomatisable logics transfers to their product. An interesting corollary of Lemma 22 is the following:

COROLLARY 23. *Let L_1, L_2 and L_3 be Kripke complete and Horn axiomatisable unimodal logics. Then*

$$L_1 \times L_2 \times L_3 \;=\; (L_1 \times L_2) \times L_3 \;=\; L_1 \times (L_2 \times L_3).$$

Unfortunately, no other general result is known about axiomatisations of two-dimensional products. In Section 3.4 we shall see several examples of pairs of finitely axiomatisable modal logics whose products are not even recursively enumerable. Such are, for instance, $\mathsf{Log}\{\langle \mathbb{N}, < \rangle\} \times \mathsf{Log}\{\langle \mathbb{N}, < \rangle\}$, $\mathbf{K4} \times \mathbf{GL.3}$ and $\mathbf{S4} \times \mathbf{Grz.3}$.

Moreover, Theorem 21 cannot be generalised even to logics whose classes of frames are definable by *universal* first-order formulas. As the following theorem shows, for many transitive logics L, the pairs of '$\mathbf{K4.3}$ and L' and '$\mathbf{Grz.3}$ and L' are not product-matching:

THEOREM 24. (i) [23] *Let L be any Kripke complete logic containing* $\mathbf{K4}$ *and having the two-element reflexive chain as its frame. Then* $\mathbf{K4.3} \times L \neq [\mathbf{K4.3}, L]$.

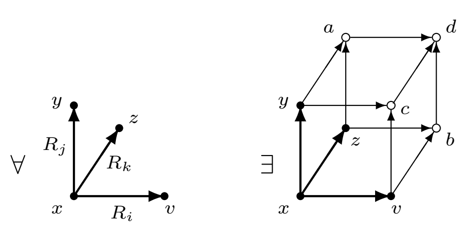
(ii) [24] *Let L_1 be any Kripke complete logic containing* $\mathbf{Grz}$ *and having the two-element reflexive chain as its frame. Let L_2 be any Kripke complete logic containing* $\mathbf{S4}$ *and having either* (a) *the two-element reflexive chain or* (b) *the two-element cluster as its frame. Then* $L_1 \times L_2 \neq [L_1, L_2]$.

There are many open questions in the area. For instance, it is not known whether such 'standard' products like $\mathbf{K4.3} \times \mathbf{K}$ or $\mathbf{K4.3} \times \mathbf{S5}$ or $\mathbf{K4.3} \times \mathbf{K4.3}$ are product-matching, or even finitely axiomatisable (they are recursively enumerable by Theorem 17). In general, no examples for pairs of logics are known that are not product-matching, but whose product is finitely axiomatisable.

Axiomatising higher dimensional product logics

Tuples of more than two modal logics are almost never product-matching. To begin with, it is straightforward to see that all n-dimensional product frames $\langle W, R_1, \dots, R_n \rangle$ satisfy the following 'cubifying' properties whenever $n \geq 3$ and $i, j, k = 1, \dots, n$ are distinct:

$$\begin{aligned}\Phi_{ijk} \;=\; & \forall x, y, z, v \in W \big(xR_iv \wedge xR_jy \wedge xR_kz \;\rightarrow\; \exists a, b, c, d \in W \\ & (vR_jc \wedge vR_kb \wedge yR_ic \wedge yR_ka \wedge zR_ib \wedge zR_ja \wedge aR_id \wedge bR_jd \wedge cR_kd)\big).\end{aligned}$$



It is not hard to see that, say, a 3-frame $\mathfrak{F}$ for $[\mathbf{K}, \mathbf{K}, \mathbf{K}]$ satisfies property Φ_{123} iff the following modal formula $\boldsymbol{cub}_{123}$ is valid in $\mathfrak{F}$ (cf. [39, 3.2.67] and [52]):

$$\begin{aligned}\boldsymbol{cub}_{123} \;=\; & [\Diamond_1(\Box_2 p_{12} \wedge \Box_3 p_{13}) \;\wedge\; \Diamond_2(\Box_1 p_{21} \wedge \Box_3 p_{23}) \;\wedge\; \Diamond_3(\Box_1 p_{31} \wedge \Box_2 p_{32}) \\ & \wedge\; \Box_1\Box_2(p_{12} \wedge p_{21} \rightarrow \Box_3 q_3) \;\wedge\; \Box_1\Box_3(p_{13} \wedge p_{31} \rightarrow \Box_2 q_2) \\ & \wedge\; \Box_2\Box_3(p_{23} \wedge p_{32} \rightarrow \Box_1 q_1)] \;\longrightarrow\; \Diamond_1\Diamond_2\Diamond_3(q_1 \wedge q_2 \wedge q_3)\,.\end{aligned}$$

Thus $\boldsymbol{cub}_{123}$ belongs to $\mathbf{K}^3$. On the other hand, Fig. 3 shows a 23-element frame for $[\mathbf{K}, \mathbf{K}, \mathbf{K}]$ (that is, a 3-frame satisfying $\boldsymbol{com}_{ij}$ and $\boldsymbol{chr}_{ij}$ for $i, j = 1, 2, 3$, $i \neq j$) that refutes $\boldsymbol{cub}_{123}$ (see again [39, 3.2.67]). So $[\mathbf{K}, \mathbf{K}, \mathbf{K}]$ and $\mathbf{K}^3$ are different.

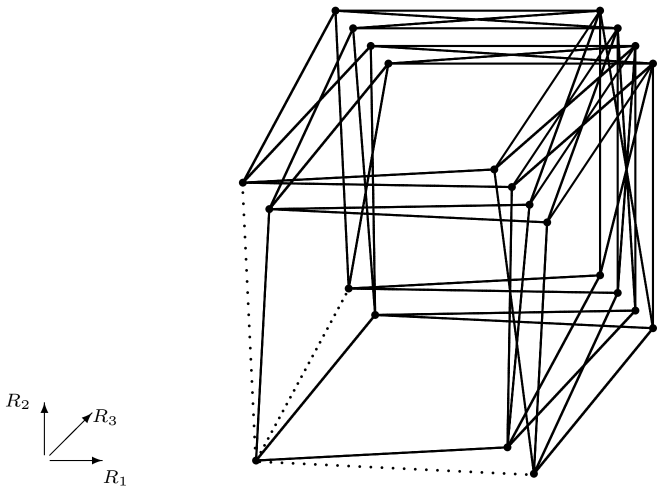


Figure 3. A frame for $[\mathbf{K}, \mathbf{K}, \mathbf{K}]$ that refutes $\boldsymbol{cub}_{123}$.

Moreover, in many cases the addition of cubifying properties does not help either. As is shown by Johnson [44] in the algebraic setting of diagonal-free cylindric algebras, $\mathbf{S5}^n$ is not finitely axiomatisable whenever $n \geq 3$. Generalisations of the cubifying properties are used in [52] to show that $\mathbf{K}^n$ is not finitely axiomatisable either for $n \geq 3$. Moreover, the following general result of [42] shows how hopeless the situation really is:

THEOREM 25. *Let $n \geq 3$ and let L be any n-modal logic such that $\mathbf{K}^n \subseteq L \subseteq \mathbf{S5}^n$. Then L is not finitely axiomatisable. Moreover, it is undecidable whether a finite n-frame is a frame for L.*

On the other hand, if frames for the component logics do not allow branching (like in the functional frames for **Alt**), then counterexamples like the above one do not work, and in fact the cubifying properties follow from the Church–Rosser properties. The following result of [24] says that any tuple of **Alt** logics is product-matching. It can be proven in a way similar to the proof of Theorem 21 above.

THEOREM 26. *For any natural number $n > 1$,* $\mathbf{Alt}^n = \overbrace{[\mathbf{Alt}, \ldots, \mathbf{Alt}]}^{n}$.

There are several interesting open questions concerning the axiomatisation of higher (≥ 3) dimensional product logics. For instance, it is not known whether logics like $\mathbf{K}^n$ or $\mathbf{S5}^n$ are axiomatisable using finitely many propositional variables, or whether $\mathbf{S5}^n$ is finitely axiomatisable over $\mathbf{K}^n$. Though logics like $\mathbf{K}^n$ or $\mathbf{S5}^n$ are known to be recursively enumerable by Theorem 17, no intuitive 'concrete' axiomatisation is known for most of them.

3.4 Decision problems and complexity of products

There are three basic approaches to establishing decidability of modal logics:

(1) Given such a logic L, one can try to prove that L has the finite model property (fmp). Even without a recursive bound on the size of the models, this can yield decidability if L is recursively enumerable, and the class of finite frames for L is recursively enumerable as well (up to isomorphism, of course). This is the case for instance if L is finitely axiomatisable.

(2) Even if a logic L does not enjoy the fmp, then one can try to show that it is characterised by some class of perhaps infinite models having a certain 'regular structure,' say, constructed from repeating finite pieces, so called 'blocks' or 'mosaics.'

(3) The third approach is to try to reduce the decision problem for L to another problem that is already known to be decidable (say, to the decision problem for another modal logic, or a suitable monadic second-order theory, or some problem about tree automata).

All three approaches have been successfully applied to uni- and multimodal logics; see e.g., [22, 12, 90]. As products of modal logics are special multimodal logics, in principle the same approaches can be applied to them as well.

As concerns (1), there is an even more tempting way. One can try to show the finite model property w.r.t. the 'intended' models, that is, those that are based on product frames. (It is important to stress that in general there are frames for product logics which are *not* product frames.)

DEFINITION 27. A modal logic L has the *product fmp* if L is characterised by the class of its finite product frames.

Note that by Proposition 14, for every product frame $\mathfrak{F} = \mathfrak{F}_1 \times \cdots \times \mathfrak{F}_n$ and product logic $L = L_1 \times \cdots \times L_n$,

$$\mathfrak{F} \models L \quad \text{iff} \quad \mathfrak{F}_i \models L_i, \text{ for all } 1 \leq i \leq n.$$

Obviously, the product fmp implies the fmp. As we shall see below, the converse does not necessarily hold.

We can enumerate the formulas that are not in a product logic L (and thereby obtain a decision algorithm for L whenever L is recursively enumerable) if

- L has the product fmp, and
- finite product frames for L are recursively enumerable (up to isomorphism).

The latter property clearly holds if L is a product of finitely axiomatisable Kripke complete logics such as **K**, **K4**, **K4.3**, **S5**, etc., so this approach looks very promising. Unfortunately, it is easy to see that most products of well-known unimodal logics lack the product fmp. Here is an example for a simple bimodal formula that 'forces' infinite product frames even for logics like **K4** $\times$ **K** or **K4** $\times$ **S5**:

$$\Box_1^+ \Diamond_2 p \wedge \Box_1^+ \Box_2 (p \to \Diamond_1 \Box_1^+ \neg p) \tag{4}$$

(here $\Box_1^+ \psi$ abbreviates $\psi \wedge \Box_1 \psi$). However, as we shall see below, two-dimensional product logics with at least one $\mathbf{S5}_n$- or $\mathbf{K}_n$-component can have the (usual, 'abstract') fmp.

Products with '$\mathbf{S5}_n$- and $\mathbf{K}_n$-like' logics are usually decidable

Filtration.

Originating in the 1940s, the *filtration method* is one of the oldest and most well-known techniques for finite model property proofs in modal logic. Here we discuss how it can be used to show the fmp of two-dimensional product logics where one component is a special kind of Horn axiomatisable logic and the other is $\mathbf{S5}_n$ or $\mathbf{K}_n$.

A *QTC-logic* is a modal logic axiomatised by a finite set of formulas where each axiom is either variable-free or of the form

$$\Box_i p \to \Box_i^j p \quad (j \geq 0) \qquad \text{or} \qquad \Diamond_k \Box_k p \to p.$$

The following theorem is due to Shehtman [76]:

THEOREM 28. *Let L_1 be a QTC-logic and L_2 be either $\mathbf{S5}_n$ or $\mathbf{K}_n$. Then $L_1 \times L_2$ has the fmp.*

As it is easy to see that every QTC-logic is Horn axiomatisable, by Theorem 21 we obtain:

THEOREM 29. *Let L_1 be a QTC-logic and L_2 be either $\mathbf{S5}_n$ or $\mathbf{K}_n$. Then $L_1 \times L_2$ is decidable.*

Proof. We illustrate the proof of Theorem 28 by showing that $\mathbf{K4} \times \mathbf{K}$ has the fmp. Suppose $\varphi \notin \mathbf{K4} \times \mathbf{K}$ for some $\mathcal{ML}_2$-formula φ. We will construct a model refuting φ that is based on a finite frame for $[\mathbf{K4}, \mathbf{K}]$. As $[\mathbf{K4}, \mathbf{K}] = \mathbf{K4} \times \mathbf{K}$ by Theorem 21, this would suffice.

As is well-known, every rooted Kripke frame is a p-morphic image of an intransitive tree. Therefore, by Propositions 9 and 12, we may assume that there exists a model $\mathfrak{M} = \langle \mathfrak{F}, \mathfrak{V} \rangle$ refuting φ and based on the product $\mathfrak{F} = \langle W, \bar{R}_1, \bar{R}_2 \rangle$ of a transitive frame and an intransitive tree of depth $md(\varphi)$. Thus, $\langle W, \bar{R}_1 \rangle$ is transitive, $\langle W, \bar{R}_2 \rangle$ is the disjoint union of intransitive trees of depth $md(\varphi)$, and $\bar{R}_1$ and $\bar{R}_2$ have the commutativity and Church–Rosser properties. For each $x \in W$, let $tree(x) = \langle W_x, \bar{R}_{2,x} \rangle$ denote the intransitive tree x belongs to.

We define an equivalence relation $\sim$ on W. For all $x, y \in W$, let $x \sim y$ iff there exists a relation $E \subseteq W_x \times W_y$ satisfying the following properties:

- xEy
- for every $u \in W_x$ there is $v \in W_y$ such that uEv,
- for every $u \in W_y$ there is $v \in W_x$ such that uEv,
- for all $u \in W_x$, $v, z \in W_y$,
  - if uEv and $v\bar{R}_{2,y}z$ then there is $z' \in W_x$ such that $u\bar{R}_{2,x}z'$ and $z'Ez$,
  - if uEv and $z\bar{R}_{2,y}v$ then there is $z' \in W_x$ such that $z\bar{R}_{2,x}u$ and $z'Ez$,
- for all $u \in W_y$, $v, z \in W_x$,
  - if uEv and $v\bar{R}_{2,x}z$ then there is $z' \in W_y$ such that $u\bar{R}_{2,y}z'$ and $z'Ez$,
  - if uEv and $z\bar{R}_{2,x}v$ then there is $z' \in W_y$ such that $z\bar{R}_{2,y}u$ and $z'Ez$,

- for all $u \in W_x$, $v \in W_y$, and propositional variables $p \in sub\,\varphi$, $u \in \mathfrak{V}(p)$ iff $v \in \mathfrak{V}(p)$.

(In other words, E should be a bisimulation between $\langle W_x, \bar{R}_{2,x}, \bar{R}_{2,x}^{-1}\rangle$ and $\langle W_y, \bar{R}_{2,y}, \bar{R}_{2,y}^{-1}\rangle$ w.r.t. $sub\,\varphi$ that connects x and y.)

Now we define a new model $\mathfrak{M}^{\sim} = \langle \mathfrak{F}^{\sim}, \mathfrak{V}^{\sim}\rangle$ based on $\mathfrak{F}^{\sim} = \left\langle W^{\sim}, \bar{R}_1^{\sim}, \bar{R}_2^{\sim}\right\rangle$ as follows:

- $W^{\sim} = \{[x] \mid x \in W\}$, where $[x]$ denotes the $\sim$-equivalence class of x;
- for all $x, y \in W$,
$$[x]\bar{R}_2^{\sim}[y] \qquad \text{iff} \qquad \exists x'\exists y'\ (x' \sim x,\ y' \sim y \text{ and } x'\bar{R}_2 y');$$
- $\bar{R}_1^{\sim}$ is the transitive closure of the relation $\bar{R}_1^{\bullet}$ defined by taking, for all $x, y \in W$,
$$[x]\bar{R}_1^{\bullet}[y] \qquad \text{iff} \qquad \exists x'\exists y'\ (x' \sim x,\ y' \sim y \text{ and } x'\bar{R}_1 y');$$
- $\mathfrak{V}^{\sim}(p) = \{[x] \mid x \in \mathfrak{V}(p)\}$, for all $p \in sub\,\varphi$, and $\mathfrak{V}^{\sim}(q) = \emptyset$, for all other propositional variables q.

We claim that

$$\mathfrak{M}^{\sim} \text{ refutes } \varphi, \text{ and} \tag{5}$$
$$\left\langle W^{\sim}, \bar{R}_1^{\sim}, \bar{R}_2^{\sim}\right\rangle \text{ is a finite frame for } [\mathbf{K4}, \mathbf{K}]. \tag{6}$$

Claim (5) follows from the fact that $\mathfrak{M}^{\sim}$ is a *filtration* of $\mathfrak{M}$ in the sense that, for all $x, y \in W$, $i = 1, 2$, the following two conditions hold:

- if $x\bar{R}_i y$ then $[x]\bar{R}_i^{\sim}[y]$,
- if $[x]\bar{R}_i^{\sim}[y]$ then $(\mathfrak{M}, y) \models \psi$ whenever $\Box_i\psi \in sub\,\varphi$ and $(\mathfrak{M}, x) \models \Box_i\psi$.

($\bar{R}_2^{\sim}$ and $\bar{R}_1^{\sim}$ are known as the *least filtration* and the *Lemmon* (or *least transitive*) *filtration*, respectively; see e.g., [12, 31].) By induction on the construction of ψ, the reader can readily check that for every $\psi \in sub\,\varphi$ and every $x \in W$,

$$(\mathfrak{M}, x) \models \psi \qquad \text{iff} \qquad (\mathfrak{M}^{\sim}, [x]) \models \psi,$$

which yields (5).

To prove (6), observe first that $\bar{R}_1^{\sim}$ is transitive by definition. Using the definition of $\sim$, it is straightforward to show that $\sim$ commutes with $\bar{R}_2$. Then this fact can be used to show that $\bar{R}_1^{\sim}$ and $\bar{R}_2^{\sim}$ commute and have the Church–Rosser property.

Finally, we show that $W^{\sim}$ is finite. Observe first that since bisimilar paths are of equal length, if $x \sim y$ then both the depth and the co-depth of x and y (in the trees *tree*(x) and *tree*(y), respectively) are the same. Moreover, for all x, y, z, if $[x]\bar{R}_2^{\sim}[y]$, $[x]\bar{R}_2^{\sim}[z]$ and $y \not\sim z$ then the submodels generated by $[y]$ and $[z]$ in $\mathfrak{M}^{\sim}$ are not isomorphic (as far as propositional variables occurring in φ are concerned). So we have

$$|W^{\sim}| \ \leq \ \sum_{k=0}^{md(\varphi)} n_k(\varphi), \tag{7}$$

where $n_0(\varphi) = 2^{|sub\,\varphi|}$ and $n_{k+1}(\varphi) = 2^{|sub\,\varphi|} \cdot 2^{n_k(\varphi)}$. ❑

Observe that the bound in (7) is non-elementary in the size of φ. In fact, it is not known whether there exists an elementary decision algorithm for $\mathbf{K}\times\mathbf{K}$ or $\mathbf{K4}\times\mathbf{K}$. Note however that products of $\mathbf{K}$ with 'richer' dynamic and temporal logics, such as $\mathbf{PDL}\times\mathbf{K}$ and $\mathbf{PTL}\times\mathbf{K}$ are known to be non-elementary; see [23]. The same applies to products with $\mathbf{S5}_n$ whenever $n \geq 2$.

On the other hand, if one component-logic is not $\mathbf{K}$ but (unimodal) $\mathbf{S5}$, then one can do better. As is shown by Gabbay and Shehtman [24], in these cases the equivalence relation $\sim$ on worlds becomes more easily 'characterisable'. Namely, for each world x in W, let

$$\Sigma(x) = \{\psi \in sub\,\varphi \mid (\mathfrak{M}, x) \models \psi\},$$

and for $x, y \in W$, put

$$x \sim y \qquad \text{iff} \qquad \Sigma(x) = \Sigma(y) \ \text{ and } \ \{\Sigma(z) \mid x\bar{R}_2 z\} = \{\Sigma(z) \mid y\bar{R}_2 z\}.$$

As now each world $[x]$ in $\mathfrak{M}^{\sim}$ is uniquely determined by the pair $\big\langle \Sigma(x), \{\Sigma(z) \mid x\bar{R}_2 z\}\big\rangle$ of sets, we have the better, double-exponential bound

$$|W^{\sim}| \ \leq \ 2^{|sub\,\varphi|} \cdot 2^{2^{|sub\,\varphi|}}$$

on the size of the filtrated model. So the filtration method yields a coN2EXPTIME decision algorithm for products of QTC-logics with $\mathbf{S5}$.

Quasimodels.

If L is Kripke complete but not a QTC-logic then $L \times \mathbf{K}_n$ and $L \times \mathbf{S5}_n$ are out of the scope of Theorem 28. Yet, many of these products can be shown to be decidable by the *quasimodel method*. This method was first developed in the series of papers [84, 85, 86, 88] on description logics with various modal and temporal operators, and then extended to products in [83, 23] and to fragments of first-order modal and temporal logics in [87, 43, 89].

The idea is to finitise the '$\mathbf{K}_n$- (or $\mathbf{S5}_n$-)bit' of the models first, then build some kind of structure that manages to keep enough information about its 'two-dimensional' character on the one hand, and can be used to prove decidability (even if it is not necessarily finite) on the other.

We fix a Kripke complete modal logic L and an $\mathcal{ML}_2$-formula φ, and define the notion of an $L \times \mathbf{K}$-*quasimodel for* φ as follows. By a *type* for φ we mean any subset $\boldsymbol{t}$ of $sub\,\varphi$ which is *Boolean-saturated* (in the sense that, for instance,

- $\psi \wedge \chi \in \boldsymbol{t}$ iff $\psi \in \boldsymbol{t}$ and $\chi \in \boldsymbol{t}$, for every $\psi \wedge \chi \in sub\,\varphi$,
- $\neg\psi \in \boldsymbol{t}$ iff $\psi \notin \boldsymbol{t}$, for every $\neg\psi \in sub\,\varphi$,

and so on for the other Boolean connectives). A *quasistate candidate* for φ is a pair $\langle\langle T, <\rangle, \boldsymbol{t}\rangle$, where $\langle T, <\rangle$ is a finite intransitive tree of depth $\leq md(\varphi)$ and $\boldsymbol{t}$ a labeling function associating with each $x \in T$ a type $\boldsymbol{t}(x)$ for φ. (So we can think of a quasistate candidate as a tree of types.) Two quasistate candidates $\langle\langle T, <\rangle, \boldsymbol{t}\rangle$ and $\langle\langle T', <'\rangle, \boldsymbol{t}'\rangle$ are called *isomorphic* if there is an isomorphism f between the trees $\langle T, <\rangle$ and $\langle T', <'\rangle$ such that $\boldsymbol{t}(x) = \boldsymbol{t}'(f(x))$, for all $x \in T$. A quasistate candidate $\langle\langle T, <\rangle, \boldsymbol{t}\rangle$ is called a *quasistate* for φ if the following conditions hold:

(qm1) ($\Diamond_2$-*saturation*) For all $x \in T$ and $\Diamond_2\psi \in sub\,\varphi$,

$$\Diamond_2\psi \in \boldsymbol{t}(x) \quad \text{iff} \quad \exists y \in T\ \big(x < y\ \wedge\ \psi \in \boldsymbol{t}(y)\big).$$

(qm1′) (*smallness*) For all $x, x_1, x_2 \in T$ such that $x < x_1$, $x < x_2$ and $x_1 \neq x_2$, the structures $\langle\langle T^{x_1}, <^{x_1}\rangle, \boldsymbol{t}^{x_1}\rangle$ and $\langle\langle T^{x_2}, <^{x_2}\rangle, \boldsymbol{t}^{x_2}\rangle$ are not isomorphic,

where $\langle T^{x_i}, <^{x_i}\rangle$ is the subtree of $\langle T, <\rangle$ generated by x_i, and $\boldsymbol{t}^{x_i}$ is the restriction of $\boldsymbol{t}$ to T^{x_i}, $i = 1, 2$. Clearly,

$$b(\varphi) = \sum_{k=0}^{md(\varphi)} n_k(\varphi) \tag{8}$$

is an upper bound for the number of different quasistates for φ (cf. (7)). The number of points in any quasistate for φ is bounded by

$$n_0(\varphi) + \sum_{k=1}^{md(\varphi)} \prod_{j=1}^{k} n_{md(\varphi)-j}(\varphi) \leq b(\varphi)^{md(\varphi)}.$$

In what follows, we assume that nonisomorphic quasistates are disjoint and that isomorphic quasistates actually coincide.

A *basic structure* of depth m for φ is a pair $\langle \mathfrak{F}, \boldsymbol{q}\rangle$ such that $\mathfrak{F} = \langle W, R\rangle$ is a frame for L and $\boldsymbol{q}$ a function associating with each $w \in W$ a quasistate $\boldsymbol{q}(w) = \langle\langle T_w, <_w\rangle, \boldsymbol{t}_w\rangle$ for φ such that the depth of each $\langle T_w, <_w\rangle$ is m.

Let $\langle \mathfrak{F}, \boldsymbol{q}\rangle$ be a basic structure for φ of depth m and let $k \leq m$. A *k-run through* $\langle \mathfrak{F}, \boldsymbol{q}\rangle$ is a function r giving for each $w \in W$ a point $r(w) \in T_w$ of depth k. (That is, a run 'goes along' the frame $\mathfrak{F}$ and chooses a (location of a) type of the same depth from each type-tree $\langle T_w, <_w\rangle$.) Given a set $\mathfrak{R}$ of runs, we denote by $\mathfrak{R}_k$ the set of all k-runs from $\mathfrak{R}$. Clearly, if $\mathfrak{R}_0$ is not empty, then it is a singleton set, with its only member r_0 being the run through the roots of the quasistates.

A run r is called *coherent* if

$$\forall w \in W\, \forall \Diamond_1\psi \in sub\,\varphi\ \Big(\exists v \in W\ \big(wRv\ \wedge\ \psi \in \boldsymbol{t}_v(r(v))\big)\ \rightarrow\ \Diamond_1\psi \in \boldsymbol{t}_w(r(w))\Big),$$

and *saturated* if

$$\forall w \in W\, \forall \Diamond_1\psi \in sub\,\varphi\ \Big(\Diamond_1\psi \in \boldsymbol{t}_w(r(w))\ \rightarrow\ \exists v \in W\ \big(wRv\ \wedge\ \psi \in \boldsymbol{t}_v(r(v))\big)\Big).$$

Finally, we say that a quadruple $\mathfrak{Q} = \langle \mathfrak{F}, \boldsymbol{q}, \mathfrak{R}, \lhd\rangle$ is an $L \times \mathbf{K}$-*quasimodel for* φ (*based on* $\mathfrak{F}$) if $\mathfrak{F}$ is a frame for L, $\langle \mathfrak{F}, \boldsymbol{q}\rangle$ is a basic structure for φ of depth $m \leq md(\varphi)$ such that

(qm2) $\exists w_0 \in W\ \varphi \in \boldsymbol{t}_{w_0}(x_0)$, where x_0 is the root of $\langle T_{w_0}, <_{w_0}\rangle$,

$\mathfrak{R}$ is a set of coherent and saturated runs through $\langle \mathfrak{F}, \boldsymbol{q}\rangle$, and $\lhd$ is a binary relation on $\mathfrak{R}$ satisfying the following conditions:

(qm3) for all $r, r' \in \mathfrak{R}$, if $r \lhd r'$ then $r(w) <_w r'(w)$ for all $w \in W$;

(qm4) $\mathfrak{R}_0 \neq \emptyset$, and for all $k < m$, $r \in \mathfrak{R}_k$, $w \in W$ and $x \in T_w$, if $r(w) <_w x$ then there is $r' \in \mathfrak{R}_{k+1}$ such that $r'(w) = x$ and $r \lhd r'$.

Now, having the notion of a quasimodel been defined, what we need is the 'quasimodel truth-lemma:'

LEMMA 30. *Given a Kripke complete modal logic L, an $\mathcal{ML}_2$-formula φ is satisfiable in a product frame $\mathfrak{F} \times \mathfrak{G}$ for $L \times \mathbf{K}$ iff there is an $L \times \mathbf{K}$-quasimodel for φ based on $\mathfrak{F}$.*

Proof. ($\Leftarrow$) Suppose $\langle \mathfrak{F}, \boldsymbol{q}, \mathfrak{R}, \lhd \rangle$ is an $L \times \mathbf{K}$-quasimodel for φ. Take the product frame $\mathfrak{F} \times \langle \mathfrak{R}, \lhd \rangle$ and define a valuation $\mathfrak{V}$ in it as follows:

$$\mathfrak{V}(p) = \{ \langle w, r \rangle \mid p \in \boldsymbol{t}_w(r(w)) \}$$

for every propositional variable p. Let $\mathfrak{M} = \langle \mathfrak{F} \times \langle \mathfrak{R}, \lhd \rangle, \mathfrak{V} \rangle$. One can show by an easy induction on the construction of $\psi \in sub\,\varphi$ that for every $\langle w, r \rangle$ in $\mathfrak{M}$ we have

$$(\mathfrak{M}, \langle w, r \rangle) \models \psi \quad \text{iff} \quad \psi \in \boldsymbol{t}_w(r(w)).$$

In view of **(qm2)** and $\mathfrak{R}_0 \neq \emptyset$ (which we have by **(qm4)**), it follows that φ is satisfied in $\mathfrak{M}$.

($\Rightarrow$) Suppose that φ is satisfied in a model $\mathfrak{M}$ based on the product $\mathfrak{F} \times \mathfrak{G}$ of frames $\mathfrak{F} = \langle W, R \rangle$ and $\mathfrak{G} = \langle \Delta, < \rangle$. By Proposition 9 (i), we may assume that $\mathfrak{G}$ is an intransitive tree of depth $m \leq md(\varphi)$ and $(\mathfrak{M}, \langle w_0, x_0 \rangle) \models \varphi$ for some $w_0 \in W$, with x_0 being the root of $\mathfrak{G}$. With every pair $\langle w, x \rangle \in W \times \Delta$ we associate the type

$$\boldsymbol{t}(w, x) = \{ \psi \in sub\,\varphi \mid (\mathfrak{M}, \langle w, x \rangle) \models \psi \}.$$

Now we have to construct a quasistate $\langle \langle T_w, <_w \rangle, \boldsymbol{t}_w \rangle$ for each $w \in W$. The obvious choice of $T_w = \Delta$, $<_w = <$ and $\boldsymbol{t}_w(x) = \boldsymbol{t}(w, x)$ does not work, because Δ can be infinite. So let us make it finite in such a way that the resulting structure still satisfies **(qm1)** and also complies with the smallness condition **(qm1′)**. Fix a $w \in W$ and define a binary relation $\sim_w$ on Δ as follows. If $x, y \in \Delta$ are of co-depth 0 (i.e., they are leaves of $\mathfrak{G}$) then

$$x \sim_w y \quad \text{iff} \quad \boldsymbol{t}(w, x) = \boldsymbol{t}(w, y).$$

For $x, y \in \Delta$ of co-depth k ($0 < k \leq md(\varphi)$), let

$$\begin{aligned} x \sim_w y \quad \text{iff} \quad \boldsymbol{t}(w, x) = \boldsymbol{t}(w, y) \quad & \text{and} \;\; \forall z \in \Delta \big(x < z \;\rightarrow\; \exists z' \in \Delta \, (y < z' \wedge z \sim_w z') \big) \\ & \text{and} \;\; \forall z \in \Delta \big(y < z \;\rightarrow\; \exists z' \in \Delta \, (x < z' \wedge z \sim_w z') \big). \end{aligned}$$

Clearly $\sim_w$ is an equivalence relation on Δ. Denote by $[x]_w$ the $\sim_w$-equivalence class of x and put

$$\begin{aligned} &\Delta_w = \{ [x]_w \mid x \in \Delta \}, \\ &[x]_w R_w [y]_w \quad \text{iff} \quad \exists y' \in [y]_w \; x < y', \\ &\boldsymbol{l}_w([x]_w) = \boldsymbol{t}(w, x). \end{aligned}$$

The structure $\langle \langle \Delta_w, R_w \rangle, \boldsymbol{l}_w \rangle$ is almost a quasistate, just $\langle \Delta_w, R_w \rangle$ is not necessarily a tree. The tree $\langle T_w, <_w \rangle$ we need can be obtained by unraveling $\langle \Delta_w, R_w \rangle$:

$$\begin{aligned} &T_w = \{ \langle [x_0]_w, \dots, [x_k]_w \rangle \mid k \leq m, \; [x_0]_w R_w [x_1]_w R_w \dots R_w [x_k]_w \}, \\ &u <_w v \quad \text{iff} \quad u = \langle [x_0]_w, \dots, [x_k]_w \rangle, \quad v = \langle [x_0]_w, \dots, [x_k]_w, [x_{k+1}]_w \rangle \\ &\qquad\qquad\quad \text{and} \;\; [x_k]_w R_w [x_{k+1}]_w. \end{aligned}$$

Finally, let $\boldsymbol{t}_w(\langle [x_0]_w, \ldots, [x_k]_w\rangle) = \boldsymbol{l}_w([x_k]_w) = \boldsymbol{t}(w, x_k)$. It is not hard to see that, for any $w \in W$, $\langle\langle T_w, <_w\rangle, \boldsymbol{t}_w\rangle$ is a quasistate for φ. Moreover, by taking $\boldsymbol{q}(w) = \langle\langle T_w, <_w\rangle, \boldsymbol{t}_w\rangle$ for each $w \in W$, we obtain a basic structure $\langle \mathfrak{F}, \boldsymbol{q}\rangle$ for φ satisfying **(qm2)**.

It remains to define appropriate runs through $\langle \mathfrak{F}, \boldsymbol{q}\rangle$. To this end, for each $k \leq m$ and each sequence $\langle x_0, \ldots, x_k\rangle$ of points in Δ such that $x_0 < \cdots < x_k$, take the map

$$r : w \mapsto \langle [x_0]_w, \ldots, [x_k]_w\rangle ,$$

and let $\mathfrak{R}$ be the set of all such maps. For $r, r' \in \mathfrak{R}$, let $r \lhd r'$ iff $r(w) <_w r'(w)$ for all $w \in W$. It is straightforward to check that $\langle \mathfrak{F}, \boldsymbol{q}, \mathfrak{R}, \lhd\rangle$ is an $L \times \mathbf{K}$-quasimodel for φ. ❑

Note that $L \times \mathbf{S5}$-quasimodels are considerably simpler than $L \times \mathbf{K}$-quasimodels: instead of trees of types it is enough to consider *sets* of types only as quasistates. Similarly to the filtration case, this results in better upper bounds on the size of the constructed structures. On the other hand, $L \times \mathbf{K}_n$- and $L \times \mathbf{S5}_n$-quasimodels (for $n \geq 2$) are similar to the above complex ones, and one even has to take into account the several different accessibility relations when defining quasistates.

Although quasistates in quasimodels are always finite, quasimodels themselves are usually infinite (since the frame $\mathfrak{F}$ can be infinite). Depending on the component logics in question, there can be several ways of using them to prove decidability of products:

- In the simplest cases one can manage to find a finite quasimodel for φ and then to construct a finite product model out of it, thereby showing that the logic has the product fmp. This can be done in the case of $\mathbf{K} \times \mathbf{K}$. Moreover, for $\mathbf{S5} \times \mathbf{S5}$ and $\mathbf{K} \times \mathbf{S5}$ the resulting product model is of exponential size (see Chapter 3 of this handbook), so these logics are decidable in coNEXPTIME.

- In some cases, it can be shown that there is a quasimodel for φ iff there exists a *finite* set $\mathcal{S}$ of *finite* 'partial' quasimodels (called *blocks* or *mosaics*) satisfying some effectively checkable conditions and that the cardinality of $\mathcal{S}$ as well as the size of each block in it do not exceed a number effectively computable from φ. The 'effectively checkable conditions' are supposed to guarantee that blocks can be used as 'small mosaic pieces' to construct the quasimodel we need.

- In some cases, the statement that a quasimodel exists can be translated into monadic second-order logic or reduced to other known decidable problems.

Here we illustrate the second and the third techniques by showing—in two different ways—that $\mathbf{K4.3} \times \mathbf{K}$ is decidable. Note that the formula (4) shows that this logic lacks the product fmp, and it is not known whether it has the fmp.

Quasimodels and mosaics.

Throughout, we fix an $\mathcal{ML}_2$-formula φ. A *block for* φ is a quadruple

$$\mathfrak{B}^{uv} = \langle \mathfrak{F}^{uv}, \boldsymbol{q}^{uv}, \mathfrak{R}^{uv}, \lhd^{uv}\rangle$$

such that

- $\mathfrak{F}^{uv} = \langle \{u, v\}, <\rangle$ is a 2-element strict linear order with $u < v$,

- $\langle \mathfrak{F}^{uv}, \boldsymbol{q}^{uv}\rangle$ is a basic structure for φ of depth m, for some $m \leq md(\varphi)$,

- $\mathfrak{R}^{uv}$ is a set of runs through $\langle \mathfrak{F}^{uv}, \boldsymbol{q}^{uv} \rangle$ such that, for all $r \in \mathfrak{R}^{uv}$ and $\Diamond_1 \psi \in sub\,\varphi$,
$$\text{if } \psi \in \boldsymbol{t}_v(r(v)) \text{ or } \Diamond_1\psi \in \boldsymbol{t}_v(r(v)) \text{ then } \Diamond_1\psi \in \boldsymbol{t}_u(r(u)),$$
- $\lhd^{uv}$ is a binary relation on $\mathfrak{R}^{uv}$ satisfying **(qm3)** and **(qm4)**.

We remind the reader that quasistates occurring in such a block are denoted by
$$\boldsymbol{q}^{uv}(u) = \langle \langle T_u, <_u \rangle, \boldsymbol{t}_u \rangle \quad \text{and} \quad \boldsymbol{q}(v)^{uv} = \langle \langle T_v, <_v \rangle, \boldsymbol{t}_v \rangle .$$

Observe that a block is almost a **K4.3** $\times$ **K**-quasimodel. The only thing missing is that its runs are (though coherent) not necessarily saturated. That is why we need an appropriate collection of blocks: By sticking them properly together, we can 'fix the defects' and converge to a real quasimodel.

To this end, we call a set $\mathcal{S}$ of blocks for φ *satisfying* if the following properties hold:

(ssb1) all blocks in $\mathcal{S}$ are of the same depth m, for some $m \leq md(\varphi)$;

(ssb2) $\mathcal{S}$ contains a block satisfying **(qm2)**;

(ssb3) for every $\mathfrak{B}^{uv}$ in $\mathcal{S}$, if $\Diamond_1\psi \in \boldsymbol{t}_v(r(v))$ for some run $r \in \mathfrak{R}^{uv}$ then there exist a block $\mathfrak{B}^{vw}$ in $\mathcal{S}$ and a sequence $\langle x_s \in T_w \mid s \in \mathfrak{R}^{uv} \rangle$ of points in T_w such that

- $\boldsymbol{q}^{uv}(v) = \boldsymbol{q}^{vw}(v)$,
- for every $s \in \mathfrak{R}^{uv}$, the function p defined by $p(v) = s(v)$, $p(w) = x_s$ is a run in $\mathfrak{R}^{vw}$,
- for all $s, s' \in \mathfrak{R}^{uv}$, if $s \lhd^{uv} s'$ then $x_s <_w x_{s'}$,
- $\psi \in \boldsymbol{t}_w(x_r)$;

(ssb4) for every block $\mathfrak{B}^{uv}$ in $\mathcal{S}$, if $\Diamond_1\psi \in \boldsymbol{t}_u(r(u))$, $\psi \notin \boldsymbol{t}_v(r(v))$ and $\Diamond_1\psi \notin \boldsymbol{t}_v(r(v))$ for some run $r \in \mathfrak{R}^{uv}$ then there are blocks $\mathfrak{B}^{uw}$ and $\mathfrak{B}^{wv}$ in $\mathcal{S}$ and a sequence $\langle x_s \in T_w \mid s \in \mathfrak{R}^{uv} \rangle$ of points in T_w such that

- $\boldsymbol{q}^{uv}(u) = \boldsymbol{q}^{uw}(u)$, $\boldsymbol{q}^{uw}(w) = \boldsymbol{q}^{wv}(w)$, $\boldsymbol{q}^{wv}(v) = \boldsymbol{q}^{uv}(v)$,
- for every $s \in \mathfrak{R}^{uv}$, the function p' defined by $p'(u) = s(u)$, $p'(w) = x_s$ is a run in $\mathfrak{R}^{uw}$, and the function p'' defined by $p''(w) = x_s$, $p''(v) = s(v)$ is a run in $\mathfrak{R}^{wv}$,
- for all $s, s' \in \mathfrak{R}^{uv}$, if $s \lhd^{uv} s'$ then $x_s <_w x_{s'}$,
- $\psi \in \boldsymbol{t}_w(x_r)$.

On the one hand, it is straightforward to see that one can effectively check whether there exists a satisfying set of blocks for φ. On the other hand, it is well-known that every rooted frame for **K4.3** is a p-morphic image of a sufficiently large strict linear order, so by Propositions 9 and 12, **K4.3** $\times$ **K** is determined by product frames whose first component is a strict linear order. As satisfiability in a single element strict linear order is trivially decidable, to establish the decidability of **K4.3** $\times$ **K**, it is enough to prove the following 'block lemma:'

LEMMA 31. *There is a* $\mathbf{K4.3} \times \mathbf{K}$*-quasimodel for* φ *based on a strict linear order with* ≥ 2 *elements iff there is a satisfying set of blocks for* φ.

Proof. The construction of a satisfying set from a quasimodel is easy. Suppose that $\mathfrak{Q} = \langle \mathfrak{F}, \boldsymbol{q}, \mathfrak{R}, \lhd \rangle$ is a quasimodel for φ, with $\mathfrak{F} = \langle W, R \rangle$ being a strict linear order with ≥ 2 elements. For all $u, v \in W$ such that uRv, define the *restriction* $\mathfrak{Q}^{uv}$ of $\mathfrak{Q}$ to the 2-element strict linear order on $\{u, v\}$ in the natural way. It is straightforward to check that these $\mathfrak{Q}^{uv}$ are blocks and that the collection $\mathcal{S}$ of them is a satisfying set.

Now we show how a quasimodel for φ can be constructed from a satisfying set $\mathcal{S}$ of blocks for φ. Starting from a block satisfying **(qm2)**, we will build a series of larger and larger quasimodel-like structures having not necessarily saturated runs. The 'defects' of these runs are 'corrected' one by one in such a way that the sequence of structures 'converges' to a quasimodel.

To begin with, we call a quadruple $\mathfrak{Q} = \langle \mathfrak{F}, \boldsymbol{q}, \mathfrak{R}, \lhd \rangle$ a *weak quasimodel for* φ if the following conditions hold:

(wq1) $\mathfrak{F} = \langle W, R \rangle$ is a finite strict linear order, $W = \{w_0, w_1, \dots, w_m\}$ for some $m > 0$, $w_0 R w_1 R \dots R w_m$, and $\langle \mathfrak{F}, \boldsymbol{q} \rangle$ is a basic structure for φ satisfying **(qm2)**;

(wq2) $\mathfrak{R}$ is a set of runs through $\langle \mathfrak{F}, \boldsymbol{q} \rangle$ such that for all $i < j \leq m$, $r \in \mathfrak{R}$ and $\Diamond_1 \psi \in sub\, \varphi$,

$$\text{if } \psi \in \boldsymbol{t}_{w_j}(r(w_j)) \text{ or } \Diamond_1 \psi \in \boldsymbol{t}_{w_j}(r(w_j)) \text{ then } \Diamond_1 \psi \in \boldsymbol{t}_{w_i}(r(w_i)),$$

(wq2′) $\lhd$ is a binary relation on $\mathfrak{R}$ satisfying **(qm4)** and such that, for all $r, s \in \mathfrak{R}$,

$$r \lhd s \qquad \text{iff} \qquad r(w_i) <_{w_i} s(w_i) \;\text{ for all } i \leq m,$$

(wq3) for every $i < m$, the restriction of $\mathfrak{Q}$ to the two-element strict linear order on $\{w_i, w_{i+1}\}$ is a block in $\mathcal{S}$.

(Note that property **(wq2′)** is a bit stronger than **(qm3)**.) Now take a triple $\langle i, r, \Diamond_1 \psi \rangle$ such that $i \leq m$, $r \in \mathfrak{R}$ and $\Diamond_1 \psi \in sub\, \varphi$. Such a triple is called a *defect in* $\mathfrak{Q}$ if $\Diamond_1 \psi \in \boldsymbol{t}_{w_i}(r(w_i))$ and for all j such that $i < j \leq m$, $\psi \notin \boldsymbol{t}_{w_j}(r(w_j))$ and $\Diamond_1 \psi \notin \boldsymbol{t}_{w_j}(r(w_j))$. If $i = m$ then such a defect is called an *end-defect*, otherwise it is a *middle-defect*.

We construct a sequence $\langle \mathfrak{Q}_n \mid n < \omega \rangle$ of weak quasimodels which 'converges' to a real quasimodel for φ. Take a block $\mathfrak{Q}_0 = \langle \mathfrak{F}_0, \boldsymbol{q}_0, \mathfrak{R}_0, \lhd_0 \rangle$ in $\mathcal{S}$ satisfying **(qm2)**. Clearly, it is a weak quasimodel for φ as well. Suppose now that we have already constructed $\mathfrak{Q}_n = \langle \mathfrak{F}_n, \boldsymbol{q}_n, \mathfrak{R}_n, \lhd_n \rangle$ such that $\mathfrak{F}_n = \langle W_n, R_n \rangle$, $W_n = \{w_0, w_1, \dots, w_m\}$ and $w_0 R_n w_1 R_n \dots R_n w_m$. If the set D_n of all defects in $\mathfrak{Q}_n$ is empty then we are done: $\mathfrak{Q}_n$ is obviously a quasimodel for φ. Otherwise, we take some $d = \langle i, r, \Diamond_1 \psi \rangle$ from D_n.

Case 1: d is a middle-defect, that is, $i < m$. By **(wq3)**, the restriction $\mathfrak{Q}^{w_i w_{i+1}}$ of $\mathfrak{Q}_n$ to the two-element strict linear order on $\{w_i, w_{i+1}\}$ is a block in $\mathcal{S}$. Choose two blocks $\mathfrak{B}^{w_i w}$ and $\mathfrak{B}^{w w_{i+1}}$ according to **(ssb4)** (with $u = w_i$ and $v = w_{i+1}$). We may assume that $w \notin W_n$. Define a basic structure $\langle \mathfrak{F}_n^d, \boldsymbol{q}_n^d \rangle$ by taking

$$\begin{aligned}
&W_n^d = W_n \cup \{w\},\\
&R_n^d = R_n \cup \{\langle w_j, w \rangle \mid j \leq i,\, w_j \in W_n\} \cup \{\langle w, w_j \rangle \mid i < j \leq m,\, w_j \in W_n\},\\
&\mathfrak{F}_n^d = \langle W_n^d, R_n^d \rangle,\\
&\boldsymbol{q}_n^d(v) = \begin{cases} \boldsymbol{q}^{w_i w}(v) = \boldsymbol{q}^{w w_{i+1}}(v), & \text{if } v = w,\\ \boldsymbol{q}_n(v), & \text{if } v \in W_n. \end{cases}
\end{aligned}$$

For all runs $s, p \in \mathfrak{R}_n$, $s' \in \mathfrak{R}^{w_i w}$, $s'' \in \mathfrak{R}^{w w_{i+1}}$, such that $s(w_i) = s'(w_i)$, $s'(w) = s''(w)$, $s''(w_{i+1}) = p(w_{i+1})$, define the function $s \cup s' \cup s'' \cup p$ on W_n^d by taking, for all $v \in W_n^d$,

$$(s \cup s' \cup s'' \cup p)(v) = \begin{cases} s(v), & \text{if } \ v = w_j,\, j \le i, \\ s'(v) = s''(v), & \text{if } \ v = w, \\ p(v), & \text{if } \ v = w_j,\, i < j \le m. \end{cases}$$

Let $\mathfrak{R}_n^d$ be the set of all such functions. Elements in $\mathfrak{R}_n^d$ of the form $s \cup s' \cup s'' \cup s$, for some $s \in \mathfrak{R}_n$, are called *extensions of* s. We call an extension $s \cup s' \cup s'' \cup s$ *good*, if $s'(w) = s''(w) = x_s$; cf. **(ssb4)**. Observe that every $s \in \mathfrak{R}_n$ has a unique good extension in $\mathfrak{R}_n^d$.

For all $s, s' \in \mathfrak{R}_n^d$, define

$$s \lhd_n^d s' \qquad \text{iff} \qquad s(v) <_v s'(v) \ \text{ for all } v \in W_n^d.$$

In other words, we 'glue together' the blocks $\mathfrak{B}^{w_i w}$ and $\mathfrak{B}^{w w_{i+1}}$ at w, and then 'insert' the resulting piece into $\mathfrak{Q}_n$ between w_i and w_{i+1}. It can be readily checked that $\mathfrak{Q}_n^d = \langle \mathfrak{F}_n^d, \boldsymbol{q}_n^d, \mathfrak{R}_n^d, \lhd_n^d \rangle$ is a weak quasimodel. Moreover, the defect d in $\mathfrak{Q}_n^d$ is 'cured' in the sense that (by **(ssb4)**) the good extension r^+ of r is such that $\psi \in \boldsymbol{t}_w(r^+(w))$.

Case 2: d is an end-defect. This case is analogous to Case 1, but we have to use **(ssb3)** instead of **(ssb4)** for 'gluing together' $\mathfrak{Q}_n$ and a block $\mathfrak{B}^{w_m w}$ at w_m.

Next we turn the remaining defects in $\mathfrak{Q}_n$ to a subset D_n^d of the set of defects in $\mathfrak{Q}_n^d$ as follows. Suppose $\langle j, s, \Diamond_1 \chi \rangle$ is a defect in D_n different from d. Let s^+ be the good extension of s and let $k = j$ if $j \le i$ and $k = j + 1$ otherwise. If $\langle k, s^+, \Diamond_1 \chi \rangle$ is a defect in $\mathfrak{Q}_n^d$ then we put it into D_n^d. Clearly, $|D_n^d| \le |D_n| - 1$. If $D_n^d \ne \emptyset$ then we take a defect $d' \in D_n^d$, construct $\mathfrak{Q}_n^{dd'}$, and so on. When all the finitely many defects in D_n are cured, we obtain a weak quasimodel $\mathfrak{Q}_{n+1}$. Note that every run $r_n \in \mathfrak{R}_n$ has a unique extension $r_{n+1} \in \mathfrak{R}_{n+1}$ obtained by taking at every step the good extension of the previous run. We call this r_{n+1} the *good extension of* r_n *in* $\mathfrak{Q}_{n+1}$.

The limit quasimodel is defined by taking $\mathfrak{F} = \langle W, R \rangle$, where $W = \bigcup_{n<\omega} W_n$, $R = \bigcup_{n<\omega} R_n$ and $\boldsymbol{q} = \bigcup_{n<\omega} \boldsymbol{q}_n$. Then clearly $\mathfrak{F}$ is a strict linear order and $\langle \mathfrak{F}, \boldsymbol{q} \rangle$ is a basic structure for φ.

For every $i < \omega$ and every sequence of runs $\langle r_n \in \mathfrak{R}_n \mid n \ge i \rangle$ such that r_{n+1} is the good extension of r_n in $\mathfrak{Q}_{n+1}$ for all $n \ge i$, take $r = \bigcup \{ r_n \mid n \ge i \}$. Let $\mathfrak{R}$ be the set of such runs. For $r = \bigcup \{ r_n \mid n \ge i \}$ and $r' = \bigcup \{ r'_n \mid n \ge j \}$ in $\mathfrak{R}$, define

$$r \lhd r' \qquad \text{iff} \qquad r_n \lhd_n r'_n, \ \text{ for all } n \ge \max(i, j).$$

We show that $\mathfrak{R}$ and $\lhd$ satisfy **(qm3)** and **(qm4)**. Indeed, suppose that r and r' are of the above form and $r \lhd r'$. Take a $w \in W$. There is an $n \ge \max(i, j)$ such that $w \in W_n$. Then $r(w) = r_n(w)$, $r'(w) = r'_n(w)$ and $r_n \lhd_n r'_n$, which implies $r(w) <_w r'(w)$ by **(wq2′)**. For **(qm4)**, suppose that $r = \bigcup \{ r_n \mid n \ge i \}$ and $r(w) <_w x$ for some $x \in T_w$. Then there is an $n \ge i$ such that $w \in W_n$, and so $r(w) = r_n(w)$. Since $\mathfrak{Q}_n$ satisfies **(qm4)**, there is an $s_n \in \mathfrak{R}_n$ such that $s_n(w) = x$ and $r_n \lhd_n s_n$. Let $s = \bigcup \{ s_m \mid m \ge n \}$, where s_{m+1} is the good extension of s_m in $\mathfrak{Q}_{m+1}$ for all $m \ge n$. Then $s(w) = s_n(w) = x$, and it is not hard to see that, by **(ssb3)**, **(ssb4)** and **(wq2′)**, $r_m \lhd_m s_m$ hold for all $m \ge n$, from which $r \lhd s$.

Finally, we show that all the runs in $\mathfrak{R}$ are coherent and saturated. Indeed, suppose that $r = \bigcup \{ r_n \mid n \ge i \}$ and $\Diamond_1 \psi \in \boldsymbol{t}_w(r(w))$ for some $w \in W$. Then there is an $n \ge i$ such

that $w \in W_n$, and so $r(w) = r_n(w)$. If $\langle w, r_n, \Diamond_1\psi\rangle$ is not a defect in $\mathfrak{Q}_n$ then there is a $v \in W_n$ such that wRv, $r_n(v) = r(v)$ and $\psi \in \boldsymbol{t}_v(r_n(v))$. And if $\langle w, r_n, \Diamond_1\psi\rangle$ is a defect in $\mathfrak{Q}_n$ then it is cured in its good extension r_{n+1} in $\mathfrak{Q}_{n+1}$: there is $v \in W_{n+1}$ such that wRv, $r_{n+1}(v) = r(v)$ and $\psi \in \boldsymbol{t}_v(r_{n+1}(v))$. Conversely, assume that $\psi \in \boldsymbol{t}_w(r(w))$ and let vRw. Then there is an $n \geq i$ such that $v, w \in W_n$. Thus $r(w) = r_n(w)$, $r(v) = r_n(v)$ and vR_nw, and so $\Diamond_1\psi \in \boldsymbol{t}_v(r(v))$ follows by **(wq2)**.

Therefore, $\mathfrak{Q} = \langle \mathfrak{F}, \boldsymbol{q}, \mathfrak{R}, \lhd\rangle$ is a **K4.3** $\times$ **K**-quasimodel for φ, as required. ❑

Observe that the decision procedure given above is again non-elementary. In fact, no elementary decision procedure is known for **K4.3** $\times$ **K**. However, as quasistates in **K4.3** $\times$ **S5**-quasimodels are of double-exponential size, one can obtain a 2EXPTIME decision algorithm for **K4.3** $\times$ **S5** as follows. Take the set of all blocks for φ (a straightforward computation shows that the cardinality of this set is also at most double-exponential in the size of φ). Eliminate iteratively those blocks for which there are no 'noneliminated' blocks satisfying **(ssb3)** and **(ssb4)**. This elimination procedure stops after at most double-exponentially many steps. Now it is not hard to show that φ is satisfiable iff the set $\mathcal{S}$ of remaining blocks contains a block satisfying **(qm2)**.

Quasimodels and reductions to monadic second-order theories.

Here we give a second proof for the decidability of **K4.3** $\times$ **K** by showing that one can translate the statement "there exists a **K4.3** $\times$ **K**-quasimodel for φ based on some strict linear order $\mathfrak{F}$" into monadic second-order logic.

Fix some $\mathcal{ML}_2$-formula φ. For every $m \leq md(\varphi)$, below we will define a monadic second order formula qm^m_φ (in the language having a binary predicate constant $<$) in such a way that the following holds:

LEMMA 32. *For any strict linear order $\mathfrak{F}$, $\mathfrak{F} \models \mathsf{qm}^m_\varphi$ for some $m \leq md(\varphi)$ iff there exists a* **K4.3** $\times$ **K**-*quasimodel for φ based on $\mathfrak{F}$.*

Though the monadic second-order theory of all strict linear orders is undecidable, we can still use this lemma to deduce decidability of **K4.3** $\times$ **K** as follows. It is not hard to see, using Theorem 16, that it is enough to consider quasimodels that are based on *countable* strict linear orders. Now for every monadic second-order formula ψ and a monadic predicate variable P not occurring in ψ, define the relativisation ψ^P of ψ to P inductively by taking $\psi^P = \psi$ for atomic ψ, $(\neg\psi)^P = \neg\psi^P$, $(\psi_1 \wedge \psi_2)^P = \psi_1^P \wedge \psi_2^P$, $(\forall x\psi)^P = \forall x(P(x) \to \psi^P)$, and $(\forall Q\psi)^P = \forall Q\psi^P$. Obviously, for any sentence ψ and any strict linear order $\mathfrak{F}$, we have $\mathfrak{F} \models \exists P(\exists x P(x) \wedge \psi^P)$ iff $\mathfrak{F}' \models \psi$ for some (nonempty) suborder $\mathfrak{F}'$ of $\mathfrak{F}$—the intended interpretation of P is the domain of $\mathfrak{F}'$. As is well-known, any countable strict linear order is a suborder of $\langle \mathbb{Q}, <\rangle$. Let λ be the first-order sentence defining the class of all strict linear orders. Then qm^m_φ (assumed not to involve P) is satisfiable in some countable strict linear order $\mathfrak{F}$ iff the monadic second-order formula

$$\exists P \left(\exists x P(x) \wedge (\lambda \wedge \mathsf{qm}^m_\varphi)^P\right)$$

holds in $\langle \mathbb{Q}, <\rangle$. As the monadic second-order theory of $\langle \mathbb{Q}, <\rangle$ is known to be decidable (see [67]), this proves the decidability of **K4.3** $\times$ **K**.

In order to define the necessary monadic second order formulas qm^m_φ for each $m \leq md(\varphi)$, we require a number of auxiliary formulas. Denote by Σ_m the set of all quasistates for φ of depth m. Given a quasistate $\boldsymbol{q} = \langle\langle T_{\boldsymbol{q}}, <_{\boldsymbol{q}}\rangle, \boldsymbol{t}_{\boldsymbol{q}}\rangle$ from Σ_m and a point a in $T_{\boldsymbol{q}}$, we denote the depth of a by $d_{\boldsymbol{q}}(a)$.

Introduce a unary predicate variable $P_{\boldsymbol{q}}$ for each $\boldsymbol{q} \in \Sigma_m$ and a unary predicate variable R^k_ψ for each $\psi \in sub\,\varphi$ and each $k \leq m$. Given a type $\boldsymbol{t}$ for φ and $k \leq m$, let

$$\chi_{\boldsymbol{t}}(\overline{R^k}(x)) = \bigwedge_{\psi \in \boldsymbol{t}} R^k_\psi(x) \wedge \bigwedge_{\substack{\psi \notin \boldsymbol{t} \\ \psi \in sub\,\varphi}} \neg R^k_\psi(x),$$

saying that the type $\boldsymbol{t}$ at point x of depth k is defined with the help of

$$\overline{R^k}(x) = \left\langle R^k_\psi(x) \mid \psi \in sub\,\varphi \right\rangle.$$

For each $k \leq m$, let

$$\begin{aligned}\mathsf{run}_0(\overline{P}, \overline{R^k}) \;=\; & \forall x \bigwedge_{\boldsymbol{q} \in \Sigma_m} \big(P_{\boldsymbol{q}}(x) \to \\ & \bigvee_{\substack{a \in T_q \\ d_q(a)=k}} \chi_{\boldsymbol{t}_q(a)}(\overline{R^k}(x))\big) \;\wedge\; \forall x \bigwedge_{\Diamond_1 \psi \in sub\,\varphi} \left[R^k_{\Diamond_1 \psi}(x) \leftrightarrow \exists y\big(x < y \wedge R^k_\psi(y)\big)\right].\end{aligned}$$

This is intended to say that $\overline{R^k}$ defines a coherent and saturated k-run through a sequence of quasistates defined with the help of $\overline{P} = \langle P_{\boldsymbol{q}} \mid \boldsymbol{q} \in \Sigma_m \rangle$.

However, we have to refine this definition in order to ensure that condition **(qm4)** holds. To this end, we define, by 'backwards' induction on k, another formula $\mathsf{run}(\overline{P}, \overline{R^k})$ as follows. If $k = m$ (that is, we are at the 'leaf-level') then take $\mathsf{run}(\overline{P}, \overline{R^m}) = \mathsf{run}_0(\overline{P}, \overline{R^m})$.

Suppose, inductively, that for $k \leq m$ we have already defined $\mathsf{run}(\overline{P}, \overline{R^k})$. Then let $\mathsf{run}(\overline{P}, \overline{R^{k-1}})$ be the following formula:

$$\begin{aligned}& \mathsf{run}_0(\overline{P}, \overline{R^{k-1}}) \wedge \\ & \forall x \bigwedge_{\boldsymbol{q} \in \Sigma_m} \bigwedge_{\substack{a \in T_q \\ d_q(a)=k-1}} \Big[P_{\boldsymbol{q}}(x) \wedge \chi_{\boldsymbol{t}_q(a)}(\overline{R^{k-1}}(x)) \to \bigwedge_{\substack{b \in T_q \\ a <_q b}} \exists_{\psi \in sub\,\varphi} R^k_\psi \Big(\mathsf{run}(\overline{P}, \overline{R^k}) \wedge \chi_{\boldsymbol{t}_q(b)}(\overline{R^k}(x)) \wedge \\ & \qquad \forall z \bigwedge_{\boldsymbol{s} \in \Sigma_m} \bigwedge_{\substack{c \in T_s \\ d_s(c)=k-1}} \big(P_{\boldsymbol{s}}(z) \wedge \chi_{\boldsymbol{t}_s(c)}(\overline{R^{k-1}}(z)) \to \bigvee_{\substack{d \in T_s \\ c <_s d}} \chi_{\boldsymbol{t}_s(d)}(\overline{R^k}(z))\big)\Big)\Big].\end{aligned}$$

Finally, we define a monadic second-order sentence qm^m_φ by taking

$$\begin{aligned}\mathsf{qm}^m_\varphi = \exists_{\boldsymbol{q} \in \Sigma_m} P_{\boldsymbol{q}} \Big[& \forall x \bigvee_{\boldsymbol{q} \in \Sigma_m} \big(P_{\boldsymbol{q}}(x) \wedge \bigwedge_{\substack{\boldsymbol{q}' \in \Sigma_m \\ \boldsymbol{q} \neq \boldsymbol{q}'}} \neg P_{\boldsymbol{q}'}(x)\big) \;\wedge \\ & \bigvee_{\substack{\boldsymbol{s} \in \Sigma_m,\, a \in T_s \\ d_s(a)=0 \\ \varphi \in \boldsymbol{t}_s(a)}} \exists x \Big(P_{\boldsymbol{s}}(x) \wedge \exists_{\psi \in sub\,\varphi} R^0_\psi \;\big(\mathsf{run}(\overline{P}, \overline{R^0}) \wedge \chi_{\boldsymbol{t}_s(a)}(\overline{R^0}(x))\big)\Big)\Big].\end{aligned}$$

Evaluated in a strict linear order $\mathfrak{F} = \langle W, < \rangle$, the first line of qm^m_φ says that the sets $P_{\boldsymbol{q}} \subseteq W$ ($\boldsymbol{q} \in \Sigma_m$) form a partition of W. By defining the map $\boldsymbol{q} : W \to \Sigma_m$ as

$$\boldsymbol{q}(w) = \boldsymbol{q} \quad \text{iff} \quad w \in P_{\boldsymbol{q}}$$

and a relation $\lhd$ on the runs by taking $r \lhd r'$ iff r is defined by $\overline{R^{k-1}}$ and r' is defined by $\overline{R^k}$ for some $k \leq m$, we obtain a quasimodel $\mathfrak{Q} = \langle \mathfrak{F}, \boldsymbol{q}, \mathfrak{R}, \lhd \rangle$ for φ: the second line of qm_φ^m states condition **(qm2)**; conditions **(qm3)** and **(qm4)** are satisfied by the definitions of $\lhd$ and the formulas $\mathsf{run}(\overline{P}, \overline{R^k})$, respectively.

Lower complexity bounds.

The following general result was obtained by Marx [59]. It is proved by reducing the NEXPTIME-complete "$n \times n$ bounded tiling problem" to the satisfiability problem of the logics in question, see Chapter 3 of this handbook:

THEOREM 33. *Let L be a Kripke complete bimodal logic between* $\mathbf{K} \times \mathbf{K}$ *and* $\mathbf{S5} \times \mathbf{S5}$. *Then L is* coNEXPTIME*-hard.*

For products of 'linear' logics with **S5** (such as $\mathsf{Log}\{\langle \mathbb{N}, < \rangle\} \times \mathbf{S5}$, $\mathsf{Log}\{\langle \mathbb{Q}, < \rangle\} \times \mathbf{S5}$, $\mathbf{K4.3} \times \mathbf{S5}$) one can obtain an EXPSPACE lower bound by reducing the "2^n corridor tiling problem" to their satisfiability problem, see [23, Theorem 6.64].

Products of 'transitive' modal logics are usually undecidable

None of the techniques for proving decidability discussed above work if we consider two-dimensional products where both component logics are determined by *transitive* frames of unbounded 'cluster-depth' (such as $\mathbf{K4} \times \mathbf{K4}$). As we shall see below, these product logics are in fact undecidable, and often lack the 'abstract' fmp.

Given a transitive frame $\mathfrak{F} = \langle W, R \rangle$, a point $x \in W$ is said to be of *cluster-depth* $n < \omega$ in $\mathfrak{F}$ if there is a path $x = x_0 R x_1 R \dots R x_n$ of points from distinct clusters in $\mathfrak{F}$ (that is, $x_{i+1} R x_i$ does not hold for any $i < n$) and there is no such path of greater length. If for every $n < \omega$ there is a path of n points from distinct clusters starting from x, then we say that x is of *infinite cluster-depth*, or x is of *cluster-depth* ∞. The *cluster-depth of* $\mathfrak{F}$ is defined to be the supremum of the cluster-depths of its points (with $n < \infty$ for all $n < \omega$). For instance, $\mathfrak{F}$ is of infinite cluster-depth if it contains points of arbitrary finite cluster-depth. By the *cluster-depth* of a bimodal frame $\langle W, R_1, R_2 \rangle$ with transitive R_1, R_2 we understand the minimal cluster-depth of $\langle W, R_1 \rangle$ and $\langle W, R_2 \rangle$.

We remind the reader that a frame $\langle W, R \rangle$ is called *Noetherian* if there is no infinite strictly ascending chain $x_0 R x_1 R x_2 R \dots$ of points from W (i.e., no R-chain such that $x_i \neq x_{i+1}$, for all $i < \omega$).

THEOREM 34. (i) [28] *Let L_1 and L_2 be Kripke complete unimodal logics containing* **K4** *and such that both L_1 and L_2 have among their frames a rooted Noetherian linear order with an infinite descending chain of distinct points. Then all bimodal logics L in the interval*

$$[L_1, L_2] \subseteq L \subseteq L_1 \times L_2$$

lack the fmp.

(ii) [26] *If L is any Kripke complete bimodal logic containing* $\mathbf{K4} \times \mathbf{K4}$ *and having product frames of arbitrarily large finite or infinite cluster-depth, then L is undecidable.*

Note that as (by Theorem 21) $\mathbf{K4} \times \mathbf{K4} = [\mathbf{K4}, \mathbf{K4}]$, it is a simple and natural example for a finitely axiomatisable but undecidable modal logic.

Below we discuss the main points of the proof of Theorem 34. For more details, consult [26].

Lack of finite model property.

We will define a bimodal formula 'forcing' infinite [**K4**, **K4**]-frames. We want to 'get rid of' the clusters first: take two fresh propositional variables h and v, and define new modal operators by setting, for every bimodal formula ψ,

$$\begin{aligned}
\bar{\Diamond}_1 \psi &= \big[h \to \Diamond_1\big(\neg h \wedge (\psi \vee \Diamond_1\psi)\big)\big] \wedge \big[\neg h \to \Diamond_1\big(h \wedge (\psi \vee \Diamond_1\psi)\big)\big],\\
\bar{\Diamond}_2 \psi &= \big[v \to \Diamond_2\big(\neg v \wedge (\psi \vee \Diamond_2\psi)\big)\big] \wedge \big[\neg v \to \Diamond_2\big(v \wedge (\psi \vee \Diamond_2\psi)\big)\big],\\
\bar{\Box}_1 \psi &= \neg\bar{\Diamond}_1 \neg\psi, \qquad \text{and} \qquad \bar{\Box}_2 \psi = \neg\bar{\Diamond}_2 \neg\psi.
\end{aligned}$$

(Similar operators were used by Spaan [77] and by Reynolds and Zakharyaschev [70].)

Now define φ_∞ to be the conjunction of the following formulas:

$$\Box_1\Box_2\big((h \vee \Diamond_2 h \to \Box_2 h) \wedge (\neg h \vee \Diamond_2\neg h \to \Box_2\neg h)\big), \tag{9}$$

$$\Box_1\Box_2\big((v \vee \Diamond_1 v \to \Box_1 v) \wedge (\neg v \vee \Diamond_1\neg v \to \Box_1\neg v)\big), \tag{10}$$

$$\bar{\Diamond}_2\,\bar{\Diamond}_1\,(\bar{\Box}_2\,\bot \wedge \bar{\Box}_1\,\bot), \tag{11}$$

$$\bar{\Box}_1\,\bar{\Box}_2\,(\bar{\Box}_2\,\bot \wedge \bar{\Box}_1\,\bot \to d), \tag{12}$$

$$\bar{\Box}_2\,\bar{\Diamond}_1\,(\neg d \wedge \bar{\Box}_1\, d), \tag{13}$$

$$\bar{\Box}_1\,\bar{\Diamond}_2\,(d \wedge \bar{\Box}_2\,\neg d), \tag{14}$$

$$\bar{\Box}_1\,\bar{\Box}_2\,(d \to \bar{\Box}_1\,\bar{\Diamond}_2\, d), \tag{15}$$

$$\bar{\Box}_1\,\bar{\Box}_2\,(\neg d \to \bar{\Box}_2\,\bar{\Diamond}_1\,\neg d). \tag{16}$$

On the one hand, it is easy to see that φ is satisfiable in a product of two rooted Noetherian linear orders each of which contains an infinite descending chain of distinct points, see Fig. 4. Note that such a frame is *infinite*.

On the other hand, we show that φ_∞ cannot be satisfied in a *finite* frame for [**K4**, **K4**]. The idea behind the proof is that, though the points 'generated by' φ_∞ do not necessarily form a nice 'backward looking $\omega \times \omega$-grid' like on Fig. 4, yet each of them can be 'characterised' by a unique pair $\langle n, m\rangle$ of natural numbers.

To this end, suppose that φ_∞ is satisfied at the root r of a model $\mathfrak{M}$ based on a (not necessarily product) frame $\mathfrak{F} = \langle W, R_1, R_2\rangle$ for [**K4**, **K4**]. Then both R_1 and R_2 are transitive, they commute and satisfy the Church–Rosser property.

We define new ($\mathfrak{M}$-dependent) binary relations $\bar{R}_1$ and $\bar{R}_2$ on W by taking, for all $x, y \in W$,

$$\begin{aligned}
x\bar{R}_1 y \quad &\text{iff} \quad \exists z \in W\ \big[xR_1 z \text{ and } \big((\mathfrak{M}, x) \models h \iff (\mathfrak{M}, z) \models \neg h\big)\\
&\qquad\quad \text{and (either } z = y \text{ or } zR_1 y)\big],\\
x\bar{R}_2 y \quad &\text{iff} \quad \exists z \in W\ \big[xR_2 z \text{ and } \big((\mathfrak{M}, x) \models v \iff (\mathfrak{M}, z) \models \neg v\big)\\
&\qquad\quad \text{and (either } z = y \text{ or } zR_2 y)\big].
\end{aligned}$$

In other words, $x\bar{R}_1 y$ iff $xR_1 y$ and either x, y are of different 'horizontal colours' in the sense that h is true in precisely one of them, or x, y are of the same h-colour (i.e., $x \models h$ iff $y \models h$), but there is a point z of different h-colour such that $xR_1 zR_1 y$. Clearly, we always have $\bar{R}_i \subseteq R_i$ ($i = 1, 2$). It is not hard to see that, by (9)–(10), $\langle W, \bar{R}_1, \bar{R}_2\rangle$ is a

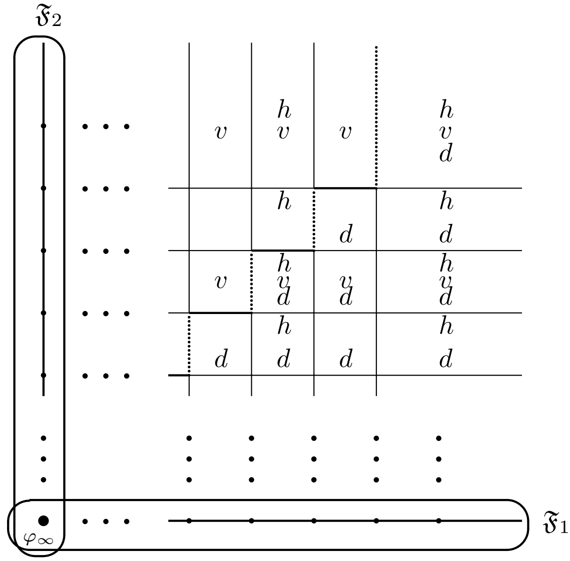


Figure 4. Satisfying φ_∞ in an infinite product frame.

(not necessarily rooted) frame for $[\mathbf{K4}, \mathbf{K4}]$, that is,

$$\text{both } \bar{R}_1 \text{ and } \bar{R}_2 \text{ are transitive,} \tag{tran}$$

$$\bar{R}_1 \text{ and } \bar{R}_2 \text{ commute, and} \tag{com}$$

$$\bar{R}_1 \text{ and } \bar{R}_2 \text{ are Church–Rosser.} \tag{chro}$$

Moreover, for all $x \in W$,

$$\begin{array}{lll}(\mathfrak{M}, x) \models \bar{\Diamond}_1 \psi & \text{iff} & \exists y \in W \ (x\bar{R}_1 y \text{ and } (\mathfrak{M}, y) \models \psi), \\ (\mathfrak{M}, x) \models \bar{\Diamond}_2 \psi & \text{iff} & \exists y \in W \ (x\bar{R}_2 y \text{ and } (\mathfrak{M}, y) \models \psi).\end{array}$$

We define inductively four infinite sequences

$$x_0, x_1, x_2, \ldots, \quad y_0, y_1, y_2, \ldots, \quad u_0, u_1, u_2, \ldots \quad \text{and} \quad v_0, v_1, v_2, \ldots$$

of points from W such that, for every $i < \omega$,

(gen1) $(\mathfrak{M}, x_i) \models d \wedge \bar{\Box}_2 \neg d$,

(gen2) $(\mathfrak{M}, y_i) \models \neg d \wedge \bar{\Box}_1 d$,

(gen3) $r\bar{R}_2 u_i$, $u_i \bar{R}_1 x_i$ and $u_i \bar{R}_1 y_i$,

(gen4) if $i > 0$ then $r\bar{R}_1 v_i$, $v_i \bar{R}_2 x_i$ and $v_i \bar{R}_2 y_{i-1}$,

see Fig. 5. (We do not claim at this point that, say, all the x_i are distinct.)

To begin with, by (11), there are u_0, x_0 such that $r\bar{R}_2 u_0 \bar{R}_1 x_0$ and

$$(\mathfrak{M}, x_0) \models \bar{\Box}_1 \bot \wedge \bar{\Box}_2 \bot. \tag{17}$$

By (12), $(\mathfrak{M}, x_0) \models d$. By (13), there is y_0 such that $u_0 \bar{R}_1 y_0$ and $(\mathfrak{M}, y_0) \models \neg d \wedge \bar{\Box}_1 d$. So **(gen1)**–**(gen3)** hold for $i = 0$.

Now suppose that, for some $n < \omega$, x_i and y_i with **(gen1)**–**(gen4)** have already been defined for all $i \leq n$. By **(gen3)** for $i = n$ and by (com), there is v_{n+1} such that $r \bar{R}_1 v_{n+1} \bar{R}_2 y_n$. So by (14), there is x_{n+1} such that $(\mathfrak{M}, x_{n+1}) \models d \wedge \bar{\Box}_2 \neg d$ and $v_{n+1} \bar{R}_2 x_{n+1}$. Now again by (com), there is u_{n+1} such that $r \bar{R}_2 u_{n+1} \bar{R}_1 x_{n+1}$. So, by (13), there is y_{n+1} such that $u_{n+1} \bar{R}_1 y_{n+1}$ and $(\mathfrak{M}, y_{n+1}) \models \neg d \wedge \bar{\Box}_1 d$, as required (see Fig. 5).

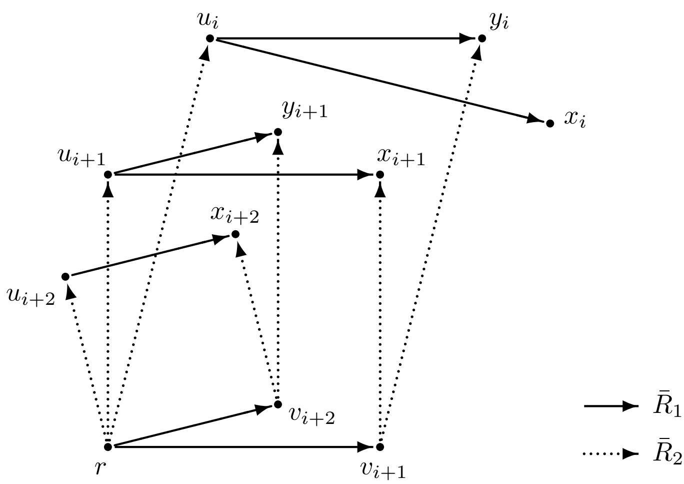


Figure 5. Generating the points x_i, y_i, u_i and v_i.

The following lemma is our basic tool in showing that all the x_n are different:

LEMMA 35. *For all $i, n < \omega$,*

(i) $(\mathfrak{M}, x_i) \models \bar{\Diamond}_1^n \top \leftrightarrow \bar{\Diamond}_2^n \top$,

(ii) $(\mathfrak{M}, y_i) \models \bar{\Diamond}_1^{n+1} \top \leftrightarrow \bar{\Diamond}_2^n \top$.

Proof. First, it is a straightforward consequence of (12), (16) and (com) that

$$\bar{\Box}_1 \bar{\Box}_2 (\neg d \to \bar{\Diamond}_1 \top) \tag{18}$$

holds in $\mathfrak{M}$. Further, it is not hard to show by induction on n that for all $n < \omega$,

$$\bar{\Box}_1 \bar{\Box}_2 (d \to \bar{\Box}_1^n \bar{\Diamond}_2^n d), \tag{19}$$

$$\bar{\Box}_1 \bar{\Box}_2 (\neg d \to \bar{\Box}_2^n \bar{\Diamond}_1^n \neg d). \tag{20}$$

are also true in $\mathfrak{M}$. Now to prove (i), suppose first that we have $(\mathfrak{M}, x_i) \models \bar{\Diamond}_1^n \top$. Then there is a point z such that $x_i \bar{R}_1^n z$. By **(gen1)**, $(\mathfrak{M}, x_i) \models d$. So, $(\mathfrak{M}, z) \models \bar{\Diamond}_2^n d$, by (19). Using (com), we find a point v such that $x_i \bar{R}_2^n v$ and $v \bar{R}_1^n u$, so $(\mathfrak{M}, x_i) \models \bar{\Diamond}_2^n \top$ follows. Conversely, suppose $(\mathfrak{M}, x_i) \models \bar{\Diamond}_2^n \top$, that is, there are points $z_1, \ldots, z_n$ such

that $x_i \bar{R}_2 z_1 \bar{R}_2 \dots \bar{R}_2 z_n$. By **(gen1)**, $(\mathfrak{M}, x_i) \models \bar{\Box}_2 \neg d$, and so $(\mathfrak{M}, z_1) \models \neg d$. Therefore, by (20) and (18), we have $(\mathfrak{M}, z_n) \models \bar{\Diamond}_1^n \top$, and then obtain $(\mathfrak{M}, x_i) \models \bar{\Diamond}_1^n \top$ using (com).

To show (ii), assume first that we have $(\mathfrak{M}, y_i) \models \bar{\Diamond}_2^n \top$. Then there is a point z such that $y_i \bar{R}_2^n z$. By **(gen2)**, $(\mathfrak{M}, y_i) \models \neg d$. So, by (20), $(\mathfrak{M}, z) \models \bar{\Diamond}_1^n \neg d$, and by (18), $(\mathfrak{M}, z) \models \bar{\Diamond}_1^{n+1} \top$. Now $(\mathfrak{M}, y_i) \models \bar{\Diamond}_1^{n+1} \top$ follows by (com). Conversely, suppose $(\mathfrak{M}, y_i) \models \bar{\Diamond}_1^{n+1} \top$, that is, there are points $z_1, \dots, z_n, z_{n+1}$ such that

$$y_i \bar{R}_1 z_1 \bar{R}_1 \dots \bar{R}_1 z_n \bar{R}_1 z_{n+1}.$$

By **(gen2)**, $(\mathfrak{M}, y_i) \models \bar{\Box}_1 d$, and so $(\mathfrak{M}, z_1) \models d$. Therefore, by (19), $(\mathfrak{M}, z_{n+1}) \models \bar{\Diamond}_2^n \top$. Finally, using (com) we obtain $(\mathfrak{M}, y_i) \models \bar{\Diamond}_2^n \top$. ❑

Now we can show that all the x_n are distinct as follows. For every formula ψ and $\Diamond \in \{\bar{\Diamond}_1, \bar{\Diamond}_2\}$, we introduce

$$\Diamond^{=\mathbf{n}}\psi \;=\; \Diamond^n \psi \wedge \Box^{n+1} \neg\psi,$$

meaning 'see ψ in n steps but not in $n+1$ steps.' Define the horizontal and vertical ranks $hr(x)$ and $vr(x)$ of a point x (in model $\mathfrak{M}$) by taking

$$hr(x) \;=\; \begin{cases} n, & \text{if } n < \omega \text{ and } (\mathfrak{M}, x) \models \bar{\Diamond}_1^{=\mathbf{n}} \top, \\ \infty, & \text{otherwise,} \end{cases}$$

$$vr(x) \;=\; \begin{cases} n, & \text{if } n < \omega \text{ and } (\mathfrak{M}, x) \models \bar{\Diamond}_2^{=\mathbf{n}} \top, \\ \infty, & \text{otherwise.} \end{cases}$$

The reader can readily check, using (com) and (chro), that if $x \bar{R}_1 y$ then $vr(x) = vr(y)$, and if $x \bar{R}_2 y$ then $hr(x) = hr(y)$.

We claim that, for all $n < \omega$,

$$vr(u_n) \;=\; n, \tag{21}$$

$$hr(v_n) \;=\; n, \tag{22}$$

$$hr(x_n) \;=\; vr(x_n) = n. \tag{23}$$

First we prove (21) by induction on n. For $n = 0$, it follows from the definition of x_0 (see (17)) and **(gen3)**. Suppose that (21) holds for some $n < \omega$. Then

$$\begin{aligned} vr(u_{n+1}) &\overset{\textbf{(gen3)}}{=} vr(x_{n+1}) \overset{\text{L.35(i)}}{=} hr(x_{n+1}) \\ &\overset{\textbf{(gen4)}}{=} hr(y_n) \overset{\text{L.35(ii)}}{=} vr(y_n) + 1 \overset{\textbf{(gen3)}}{=} vr(u_n) + 1 \overset{\text{(IH)}}{=} n + 1. \end{aligned}$$

Now (22) and (23) follow from (21) and

$$hr(v_n) \overset{\textbf{(gen4)}}{=} hr(x_n) \overset{\text{L.35(i)}}{=} vr(x_n) \overset{\textbf{(gen3)}}{=} vr(u_n).$$

Undecidability.

We discuss first how the 'diagonal points' x_n (with finite rank $hr(x_n) = vr(x_n) = n$) can be used not only to show the lack of fmp, but also to encode arbitrarily large finite parts of the '$\omega \times \omega$-grid' in frames for $[\mathbf{K4}, \mathbf{K4}]$. The enumeration of the points of $\omega \times \omega$

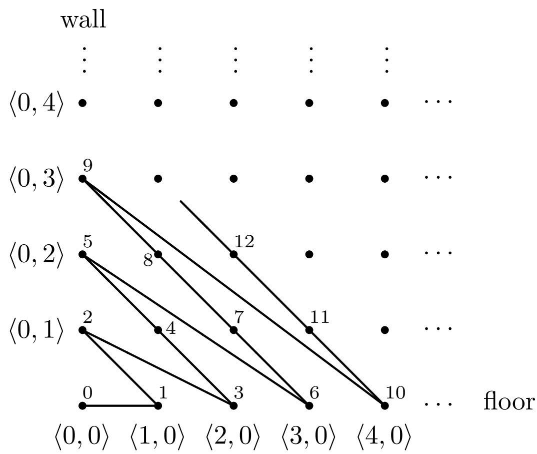


Figure 6. The enumeration *pair*.

we use below has been introduced in several papers dealing with undecidable multimodal logics; see, e.g., [36, 60, 70]. (Note that in all these cases either the language had *next-time operators* or all the frames were *linear*, neither is the case now.)

Let $pair : \omega \to \omega \times \omega$ be the function defined recursively by taking:

- $pair(0) = \langle 0, 0\rangle$,
- if $pair(n) = \langle 0, j\rangle$ then $pair(n+1) = \langle j+1, 0\rangle$,
- otherwise, if $pair(n) = \langle i+1, j\rangle$ then $pair(n+1) = \langle i, j+1\rangle$;

see Fig. 6. It is easy to see that *pair* is one-one and onto. Let $\sharp : \omega \times \omega \to \omega$ denote the inverse of the function *pair*. If $pair(n)$ is not on the wall (that is, the first coordinate of $pair(n)$ is different from 0) then define $left_n$ to be the $\sharp$ of the left neighbour of $pair(n)$. The reader can readily check the following important properties of these functions, for all $n > 0$:

(t1) If neither $pair(n)$ nor $pair(n-1)$ are on the wall then $left_n = left_{n-1} + 1$.

(t2) If $n > 1$ and $pair(n)$ is not on the wall, but $pair(n-1)$ is on the wall, then $n > 2$, $pair(n-2)$ is not on the wall, and $left_n = left_{n-2} + 1$.

(t3) $pair(n)$ is on the wall iff $pair(left_{n-1})$ is on the wall.

(t4) Either $pair(n)$ or $pair(n-1)$ is not on the wall.

We will require the following propositional variables:

- grid (marking the points of the grid),
- left (a pointer from n to $left_n$ when $pair(n)$ is not on the wall),
- wall (marking the wall, i.e., the pairs of the form $\langle 0, n\rangle$).

Let φ_{grid} be the conjunction of (9), (10) and the following formulas:

$$\bar{\Box}_1\,\bar{\Box}_2\left(\bar{\Box}_1\bot \rightarrow (\mathsf{grid} \leftrightarrow \bar{\Box}_2\bot)\right),$$
$$\bar{\Box}_1\,\bar{\Box}_2\,(\bar{\Box}_1\bot \wedge \mathsf{grid} \rightarrow \mathsf{wall}),$$
$$\bar{\Box}_1\,\bar{\Box}_2\,(\mathsf{wall} \rightarrow \mathsf{grid}),$$
$$\bar{\Box}_1\,\bar{\Box}_2\left(\bar{\Diamond}_1\mathsf{wall} \rightarrow \bar{\Box}_1(\mathsf{grid} \rightarrow \mathsf{wall})\right),$$
$$\bar{\Box}_1\,\bar{\Box}_2\left(\bar{\Diamond}_1\top \rightarrow (\mathsf{grid} \leftrightarrow \bar{\Diamond}_2^{=\mathbf{1}}\,\bar{\Diamond}_1^{=\mathbf{1}}\,\mathsf{grid})\right),$$
$$\bar{\Box}_1\,\bar{\Box}_2\left(\mathsf{grid} \wedge \bar{\Diamond}_1\top \rightarrow \left(\mathsf{wall} \leftrightarrow \bar{\Diamond}_2(\bar{\Diamond}_1^{=\mathbf{1}}\,\mathsf{left} \wedge \bar{\Diamond}_1\mathsf{wall})\right)\right),$$
$$\bar{\Box}_1\,\bar{\Box}_2\Big[\mathsf{left} \leftrightarrow \Big((\bar{\Diamond}_1^{=\mathbf{1}}\top \wedge \bar{\Box}_2\bot) \vee \left(\bar{\Diamond}_2(\bar{\Diamond}_1^{=\mathbf{2}}\,\mathsf{left} \wedge \bar{\Diamond}_1\mathsf{wall}) \wedge \bar{\Diamond}_2^{=\mathbf{1}}\,\bar{\Diamond}_1^{=\mathbf{2}}\,\mathsf{left}\right)$$
$$\vee \left(\bar{\Diamond}_2(\bar{\Diamond}_1^{=\mathbf{1}}\,\mathsf{left} \wedge \neg\bar{\Diamond}_1\mathsf{wall}) \wedge \bar{\Diamond}_2^{=\mathbf{1}}\,\bar{\Diamond}_1^{=\mathbf{1}}\,\mathsf{left}\right)\Big)\Big].$$

The following lemma, showing that φ_{grid} 'forces' the $\omega \times \omega$-grid onto 'diagonal points of finite rank', is proved in [26]:

LEMMA 36. *Suppose that $\mathfrak{M}$ is a model based on a rooted frame $\mathfrak{F} = \langle W, R_1, R_2\rangle$ for* $[\mathbf{K4}, \mathbf{K4}]$. *If $(\mathfrak{M}, r) \models \varphi_{grid}$ then the following hold, for all $n, m < \omega$ and all $x \in W$ such that $hr(x) = n$ and $vr(x) = m$:*

(i) $(\mathfrak{M}, x) \models \mathsf{grid}$ *iff* $n = m$,

(ii) $(\mathfrak{M}, x) \models \bar{\Diamond}_1^{=\mathbf{1}}\,\mathsf{left}$ *iff* $n > 0$, *$pair(n-1)$ is not on the wall and $m = left_{n-1}$,*

(iii) $(\mathfrak{M}, x) \models \mathsf{wall}$ *iff* $n = m$ *and $pair(n)$ is on the wall,*

(iv) $(\mathfrak{M}, x) \models \mathsf{left}$ *iff $pair(n)$ is not on the wall and $m = left_n$.*

Various undecidable problems can be 'represented' on the $\omega \times \omega$-grid, say, versions of the halting problems for Turing machines, register machines, etc., Post's correspondence problem, as well as infinite tiling problems.

Here we show as an example for reducing an undecidable tiling problem to the satisfiability problem for logics that (i) contain $[\mathbf{K4}, \mathbf{K4}]$ and (ii) have among their frames a product of two rooted Noetherian linear orders each of which contains an infinite descending chain of distinct points. (For other similar logics slight modifications of the proof might be necessary, see [26] for a general argument.)

A *tile type* is a 4-tuple of colours

$$t = \langle left(t), right(t), up(t), down(t)\rangle\,.$$

For a *finite* set T of tile types and a subset $X \subseteq \omega \times \omega$, we say that T *tiles* X if there exists a function (called a *tiling*) $\boldsymbol{\tau}$ from X to T such that, for all $\langle i, j\rangle \in X$,

- if $\langle i, j+1\rangle \in X$ then $up(\boldsymbol{\tau}(i,j)) = down(\boldsymbol{\tau}(i, j+1))$ and
- if $\langle i+1, j\rangle \in X$ then $right(\boldsymbol{\tau}(i,j)) = left(\boldsymbol{\tau}(i+1, j))$.

The following '*$\omega \times \omega$-tiling problem*' is undecidable (see [81, 9]): given a finite set T of tile types, decide whether T can tile $\omega \times \omega$.

Given a finite set T of tile types, we introduce a propositional variable t, for every $t \in T$. Let φ_T be the conjunction of the following formulas:

$$\bar{\Box}_1 \bar{\Box}_2 \,(\mathsf{grid} \leftrightarrow \bigvee_{t \in T} t),$$

$$\bar{\Box}_1 \bar{\Box}_2 \bigwedge_{t \neq t' \in T} \neg(t \wedge t'),$$

$$\bar{\Box}_1 \bar{\Box}_2 \bigwedge_{\substack{t,t' \in T \\ up(t') \neq down(t)}} \big(t \to \bar{\Box}_2 \,(\bar{\Diamond}_1^{=1} \,\mathsf{left} \to \neg \bar{\Diamond}_1 t')\big),$$

$$\bar{\Box}_1 \bar{\Box}_2 \bigwedge_{\substack{t,t' \in T \\ right(t') \neq left(t)}} \big(t \to \bar{\Box}_2 \,(\mathsf{left} \to \neg \bar{\Diamond}_1 t')\big).$$

For every $n < \omega$, let

$$plane_n = \{\langle i, j\rangle \mid \sharp(i,j) \leq n\}.$$

If formulas (9) and (10) are satisfied in a model $\mathfrak{M}$ based on a frame for $[\mathbf{K4}, \mathbf{K4}]$, then for all numbers $a, b < \omega$ and $x \in W$ with $hr(x) = a$ and $vr(x) = b$, there exists what we call a *perfect* $a \times b$*-rectangle starting at* x, that is, there are points $x_{i,j}$ (for $i \leq a$, $j \leq b$) such that

- $x = x_{a,b}$,
- $hr(x_{i,j}) = i$ and $vr(x_{i,j}) = j$,
- $x_{i,j} \bar{R}_1 x_{k,j}$ for $i > k$, and $x_{i,j} \bar{R}_2 x_{i,k}$ for $j > k$.

(Indeed, given x, take an a-long $\bar{R}_1$-path and a b-long $\bar{R}_2$-path starting from x, and then 'close them' under the Church-Rosser property.)

Now a straightforward induction on n shows the following:

LEMMA 37. *Let* $\mathfrak{M}$ *be a model that is based on a frame for* $[\mathbf{K4}, \mathbf{K4}]$ *with root* r *and suppose that* $(\mathfrak{M}, r) \models \varphi_{grid} \wedge \varphi_T$. *Then, for every* $n < \omega$, *every* $x \in W$ *such that* $hr(x) = vr(x) = n$, *and every perfect* $n \times n$*-rectangle* $x_{i,j}$ $(i \leq n,\ j \leq n)$ *starting at* x, *the function* $\tau : plane_n \to T$ *defined by*

$$\tau(i,j) = t \qquad \textit{iff} \qquad (\mathfrak{M}, x_{\sharp(i,j),\sharp(i,j)}) \models t$$

is a tiling of $plane_n$.

Now, using Lemma 37, it is straightforward to show that if $\varphi_\infty \wedge \varphi_{grid} \wedge \varphi_T$ is satisfiable in a frame for $[\mathbf{K4}, \mathbf{K4}]$ then T tiles $plane_n$, for all $n < \omega$. A standard compactness argument (or König's lemma) shows that if a given finite set T of tile types tiles $plane_n$ for every $n < \omega$, then it actually tiles the whole $\omega \times \omega$-grid. On the other hand, it is easy to see that if T tiles $\omega \times \omega$ then $\varphi_\infty \wedge \varphi_{grid} \wedge \varphi_T$ is satisfiable in a product of two rooted Noetherian linear orders each of which contains an infinite descending chain of distinct points.

With the help of some additional 'machinery', one can even reduce 'stronger' undecidable statements (like recurrent Turing machine and tiling problems) to the satisfiability

problem for certain products of 'transitive' logics. For instance, the following is shown in [26]:

THEOREM 38. *Let L_1 be any logic from the list*

$$\mathbf{K4},\ \mathbf{K4.1},\ \mathbf{K4.2},\ \mathbf{K4.3},\ \mathbf{S4},\ \mathbf{S4.1},\ \mathbf{S4.2},\ \mathbf{S4.3},$$
$$\mathbf{GL},\ \mathbf{GL.3},\ \mathbf{Grz},\ \mathbf{Grz.3},\ \mathsf{Log}\{\langle\omega,<\rangle\},\ \mathsf{Log}\{\langle\omega,\leq\rangle\},$$

and L_2 be any of

$$\mathsf{Log}\{\langle\omega,<\rangle\},\ \mathsf{Log}\{\langle\omega,\leq\rangle\},\ \mathbf{GL.3},\ \mathbf{Grz.3}.$$

Then any Kripke complete bimodal logic L in the interval

$$[L_1,L_2]\ \subseteq\ L\ \subseteq\ L_1\times L_2$$

is Π^1_1-hard.

We also obtain the following interesting corollary. As the commutator of two recursively axiomatisable logics is recursively axiomatisable by definition, Theorem 38 yields a number of *Kripke incomplete* commutators of Kripke complete and finitely axiomatisable logics:

COROLLARY 39. *Let L_1 and L_2 be like in Theorem 38. Then the commutator $[L_1,L_2]$ is Kripke incomplete.*

Higher dimensional decidable and undecidable products

Products of more than two modal logics are often undecidable and lack the fmp. Let us first discuss some exceptions.

It is not hard to see that any product $L_1\times\dots\times L_n$ of **Alt** and **K** logics has the *finite depth property*, that is, it is determined by some class of frames of finite depth. Indeed, suppose $\varphi\notin L_1\times\dots\times L_n$ for some $\mathcal{ML}_n$-formula φ. Then there are rooted frames $\mathfrak{F}_i$, $i=1,\dots,n$, such that $\mathfrak{F}_i\models L_i$ and φ is refuted at the root of $\mathfrak{F}_1\times\dots\times\mathfrak{F}_n$. By a standard unravelling argument, for each $i=1,\dots,n$, there is an intransitive tree $\mathfrak{T}_i$ and a p-morphism h_i from $\mathfrak{T}_i$ onto $\mathfrak{F}_i$. So we always have $\mathfrak{T}_i\models L_i$. Note that if $L_i=\mathbf{Alt}$ then the unravelling $\mathfrak{T}_i$ of $\mathfrak{F}_i$ is just a chain of irreflexive points. It is straightforward to check that the function h defined by

$$h(x_1,\dots,x_n)=\langle h_1(x_1),\dots,h_n(x_n)\rangle$$

is a p-morphism from $\mathfrak{T}_1\times\dots\times\mathfrak{T}_n$ onto $\mathfrak{F}_1\times\dots\times\mathfrak{F}_n$ (cf. Proposition 9). Now we prune all the trees $\mathfrak{T}_i$ down to the modal depth $md(\varphi)$ of φ. Clearly, the resulting product frame $\mathfrak{T}_1^-\times\dots\times\mathfrak{T}_n^-$ is of depth $n\cdot md(\varphi)$ and it is a frame for $L_1\times\dots\times L_n$. A straightforward induction on the stucture of φ shows that it refutes φ at its root.

As one can prove by a standard filtration argument that if an n-modal logic L has the finite depth property, then it has the fmp as well, we obtain the following theorem of Gabbay and Shehtman [24]:

THEOREM 40. *Any product of* **Alt** *and* **K** *logics has the fmp. In particular,* $\mathbf{K}^n$ *and* $\mathbf{Alt}^n$ *have the fmp, for any natural number $n\geq 2$.*

As a consequence of this theorem and Theorem 26 we have:

THEOREM 41. $\mathbf{Alt}^n$ *is decidable, for any natural number* $n \geq 2$.

In fact, it can be shown that $\mathbf{Alt}^n$ has the polynomial *product* fmp and it is coNP-complete.

On the other hand, the logic $\mathbf{K}^n$ is not so simple. Though it has the fmp, one cannot use it for a decision algorithm, as $\mathbf{K}^n$ is not only not finitely axiomatisable, but it is undecidable whether a finite n-frame is a frame for it (cf. Theorem 25). In fact, the following general result is shown in [42]:

THEOREM 42. *Let* $n \geq 3$ *and let* L *be any* n*-modal logic such that* $\mathbf{K}^n \subseteq L \subseteq \mathbf{S5}^n$. *Then* L *is undecidable and lacks the product fmp.*

The proof of this theorem (and that of Theorem 25) uses a reduction of a deep result of Hirsch and Hodkinson [40] saying that representability is undecidable for finite relation algebras.

Note that, unlike $\mathbf{K}^n$, logics like $\mathbf{K4}^n$ and $\mathbf{S5}^n$ do not even have the fmp (for $\mathbf{K4}^n$ this follows from Theorem 34, and for $\mathbf{S5}^n$ this was shown in [53]). The undecidability of $\mathbf{S5}^n$ was first shown by Maddux [58] in the algebraic framework of diagonal-free cylindric algebras. He used a reduction of the word problem of semigroups to prove the following general result:

THEOREM 43. *Any* n*-modal logic* L *in the interval*

$$[\mathbf{S5}, \mathbf{S5}, \ldots, \mathbf{S5}] \subseteq L \subseteq \mathbf{S5}^n$$

is undecidable whenever $n \geq 3$.

Another proof via the connection with first-order logic (see Section 3.2) that uses a reduction of the $\omega \times \omega$ tiling problem can be found in [23] (see also [47] for possible generalisations).

4 BETWEEN FUSIONS AND PRODUCTS

A natural idea for reducing the strong interaction between modal operators of product logics is to consider logics determined by (not necessarily generated) *subframes* of product frames. Worlds are still tuples, the relations still act coordinate-wise, but not all tuples of the Cartesian product are present, and so the commutativity and Church–Rosser properties do not necessarily hold. This kind of restriction on the 'domains' of modal operators is similar to 'relativisations' of the quantifiers in first-order logic and algebraic logic, where it indeed results in improving the bad algorithmic behaviour, cf. [63, 61].

This idea gives rise to the following combinations of logics. First, we choose a class of 'desirable' subframes of product frames. This can be any class: the class of all such subframes, the so-called 'locally cubic' frames, frames that 'expand' along one of the coordinates (see below for precise definitions), a class of frames satisfying some (modal or first-order) formulas, etc. Having chosen such a class $\mathcal{K}$, we then take the logic determined by those subframes of the appropriate product frames that belong to $\mathcal{K}$. Thus, each choice of $\mathcal{K}$ defines a new combination operator on logics:

DEFINITION 44. Let $n > 1$ be a natural number and $\mathcal{K}$ a class of subframes of n-ary product frames. Given Kripke complete (uni)modal logics $L_1, \ldots, L_n$ formulated in the language having $\Box_i$ $(i = 1, \ldots, n)$, the $\mathcal{K}$*-relativised product* $(L_1 \times \cdots \times L_n)^{\mathcal{K}}$ of $L_1, \ldots, L_n$

is defined by taking

$$(L_1 \times \cdots \times L_n)^{\mathcal{K}} = \mathsf{Log}\{\mathfrak{G} \in \mathcal{K} \mid \mathfrak{G} \subseteq \mathfrak{F}_1 \times \cdots \times \mathfrak{F}_n \text{ for some } \mathfrak{F}_i \in \mathsf{Fr}L_i,\ i = 1, \ldots, n\}.$$

Observe that if we choose $\mathcal{K}$ to be the class of all product frames $\mathfrak{F}_1 \times \cdots \times \mathfrak{F}_n$ such that $\mathfrak{F}_i \in \mathsf{Fr}L_i$, then the $\mathcal{K}$-relativised product of the L_i is just their usual product.

We discuss here in detail two kinds of 'relativised product' operators: arbitrary and expanding relativisations.

Arbitrary relativisations.

We begin by considering the combination operator determined by the class SF_n of *all* subframes of n-ary product frames. SF_n-relativised products of logics will be called *arbitrarily relativised products*. Since SF_n contains frames that do not satisfy commutativity and/or Church–Rosser properties,

$$(L_1 \times \cdots \times L_n)^{\mathsf{SF}_n} \subsetneq L_1 \times \cdots \times L_n.$$

On the other hand, unlike product logics, arbitrarily relativised products do not necessarily contain the fusion of their components. For example, consider the minimal deontic logic $\mathbf{D}$, which is known to be determined by the class of *serial* frames. The formula $\Diamond_2 \top$ clearly belongs to $\mathbf{K} \otimes \mathbf{D}$, but is refuted in any *finite* subframe of, say, $\langle \omega, < \rangle \times \langle \omega, < \rangle$, and so $\Diamond_2 \top \notin (\mathbf{K} \times \mathbf{D})^{\mathsf{SF}_2}$.

However, for a large class of natural logics, arbitrarily relativised products do contain the fusions. A Kripke complete modal logic L is called a *subframe logic* if the class of Kripke frames for L is closed under taking (not necessarily generated) subframes (see Chapter 7 of this handbook). Typical examples of subframe logics are modal logics whose classes of Kripke frames are definable by universal first-order formulas, such as $\mathbf{K}$, $\mathbf{Alt}$, $\mathbf{T}$, $\mathbf{K4}$, $\mathbf{S4}$, $\mathbf{S5}$, $\mathbf{K5}$, $\mathbf{K45}$, $\mathbf{S4.3}$, and $\mathbf{K4.3}$. Note, however, that subframe logics like $\mathbf{GL}$, $\mathbf{GL.3}$, $\mathbf{Grz}$ are not first-order definable. It is not hard to see the following:

PROPOSITION 45. *If $L_1, \ldots, L_n$ are subframe logics, then*

$$L_1 \otimes \cdots \otimes L_n \subseteq (L_1 \times \cdots \times L_n)^{\mathsf{SF}_n}.$$

As the following result of [54] shows, for many standard subframe logics the converse inclusion holds as well. Thus in several cases 'arbitrary relativisation' can be regarded as a 'many-dimensional' semantical characterisation of fusions of these logics.

THEOREM 46. *Let $L_i \in \{\mathbf{K}, \mathbf{T}, \mathbf{K4}, \mathbf{S4}, \mathbf{S5}, \mathbf{S4.3}\}$, for $i = 1, \ldots, n$. Then*

$$(L_1 \times \cdots \times L_n)^{\mathsf{SF}_n} = L_1 \otimes \cdots \otimes L_n.$$

The proof is based on the following lemma that can be proved by constructing the necessary p-morphism in a step-by-step manner, see [54]:

LEMMA 47. *Suppose that $L_i \in \{\mathbf{K}, \mathbf{T}, \mathbf{K4}, \mathbf{S4}, \mathbf{S5}, \mathbf{S4.3}\}$, $i = 1, \ldots, n$, and let $\mathfrak{G} = \langle W, S_1, \ldots, S_n \rangle$ be a countable rooted n-frame such that $\langle W, S_i \rangle \models L_i$ for all $i = 1, \ldots, n$. Then $\mathfrak{G}$ is a p-morphic image of a subframe of some product frame for $L_1 \times \cdots \times L_n$.*

It is not clear how far Theorem 46 can be generalised. It definitely does not hold for all subframe logics, not even for those of them that are characterised by universally

definable classes of frames. Take, for instance, the logic **K5** that is characterised by the class of Euclidean frames, i.e., frames $\langle W, R\rangle$ satisfying the universal (Horn) sentence

$$\forall x \forall y \forall u \left(R(u,x) \wedge R(u,y) \to R(x,y)\right).$$

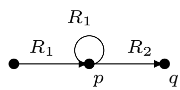
In particular, frames for **K5** have the property

$$\forall x \forall u \left(R(u,x) \to R(x,x)\right).$$

Now consider the formula

$$\varphi \;=\; \Diamond_1\big(p \wedge \Diamond_2(q \wedge \neg p)\big) \wedge \Box_1\Box_2(q \to \neg\Diamond_1 q).$$

It is clearly satisfiable in the following frame for $\mathbf{K5} \otimes \mathbf{K}$:



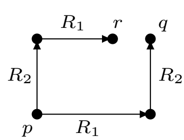
On the other hand, it is not hard to see that φ is not satisfiable in any subframe of a product frame for $\mathbf{K5} \times \mathbf{K}$. Therefore,

$$\mathbf{K5} \otimes \mathbf{K} \;\subsetneq\; (\mathbf{K5} \times \mathbf{K})^{\mathsf{SF}_2} \;\subsetneq\; \mathbf{K5} \times \mathbf{K}.$$

Other kinds of logics for which Theorem 46 does not hold are those having frames with a finite bound on their branching like **Alt**. The formula

$$\psi \;=\; p \wedge \Diamond_1\big(\neg p \wedge \Diamond_2 q\big) \wedge \Diamond_2\big(\neg p \wedge \Diamond_1 r\big) \wedge \Box_1\Box_2(q \to \neg r)$$

is clearly satisfiable in the $\mathbf{Alt} \otimes \mathbf{Alt}$-frame



On the other hand, it should be clear that ψ is not satisfiable in any subframe of a frame for $\mathbf{Alt} \times \mathbf{Alt}$. Thus,

$$\mathbf{Alt} \otimes \mathbf{Alt} \;\subsetneq\; (\mathbf{Alt} \times \mathbf{Alt})^{\mathsf{SF}_2} \;\subsetneq\; \mathbf{Alt} \times \mathbf{Alt}.$$

Expanding relativisations.

First-order modal and intuitionistic logics motivate our next combination operator. (To keep the notation simple, we concentrate on the $n = 2$ case only.)

DEFINITION 48. A 2-frame $\mathfrak{G} = \langle W, S_1, S_2\rangle$ is called an *expanding relativised product frame* if there exist frames $\mathfrak{F}_1 = \langle U_1, R_1\rangle$ and $\mathfrak{F}_2 = \langle U_2, R_2\rangle$ such that

- $\mathfrak{G}$ is a subframe of $\mathfrak{F}_1 \times \mathfrak{F}_2$, and
- for all $\langle w_1, w_2\rangle \in W$ and $u \in U_1$, if $w_1 R_1 u$ then $\langle u, w_2\rangle \in W$.

Define EX to be the class of all expanding relativised product frames. Given Kripke complete unimodal logics L_1 and L_2, the logic $(L_1 \times L_2)^{\mathsf{EX}}$ is called the *expanding relativised product* of L_1 and L_2.

Similarly to Proposition 45, if both L_1 and L_2 are subframe logics then $(L_1 \times L_2)^{\mathsf{EX}}$ is a (conservative) extension of both L_1 and L_2. Moreover, it is easy to see that every expanding relativised product frame satisfies the *left* commutativity and Church–Rosser properties (but not necessarily right commutativity). Now define the *e-commutator*

$$[L_1, L_2]^{\mathsf{EX}}$$

of L_1 *and* L_2 as the smallest bimodal logic containing L_1, L_2 and the axioms $\boldsymbol{com}^{\boldsymbol{l}}_{12}$ and $\boldsymbol{chr}_{12}$. Then clearly we have

$$[L_1, L_2]^{\mathsf{EX}} \subseteq (L_1 \times L_2)^{\mathsf{EX}}.$$

Similarly to Theorem 21, it can be shown that for some cases the e-commutator and the expanding relativised product coincide:

THEOREM 49. *Suppose* L_1 *and* L_2 *are Kripke complete unimodal logics such that* L_1 *is one of* **K**, **T**, **K4**, **S4**, **S5** *and* L_2 *is Horn axiomatisable. Then*

$$(L_1 \times L_2)^{\mathsf{EX}} = [L_1, L_2]^{\mathsf{EX}}.$$

No other general axiomatisation result for expanding relativised products is known.

As concerns decision problems, it is not hard to see that expanding relativised products are reducible to products. Indeed, let φ be an $\mathcal{ML}_2$-formula and e a propositional variable which does not occur in φ. Define by induction on the construction of φ an $\mathcal{ML}_2$-formula φ^e as follows:

$$\begin{array}{rclcrcl} p^e & = & p \quad (p \text{ a propositional variable}), & & & & \\ (\psi \wedge \chi)^e & = & \psi^e \wedge \chi^e, & & (\neg\psi)^e & = & \neg\psi^e, \\ (\Box_1\psi)^e & = & \Box_1\psi^e, & & (\Box_2\psi)^e & = & \Box_2(e \to \psi^e). \end{array}$$

By a structural induction on φ, one can easily prove the following:

PROPOSITION 50. *For all Kripke complete unimodal logics* L_1 *and* L_2 *and all* $\mathcal{ML}_2$*-formulas* φ*,*

$$\varphi \in (L_1 \times L_2)^{\mathsf{EX}} \quad \textit{iff} \quad \Big(e \wedge \Box_1^{\leq md(\varphi)}\Box_2^{\leq md(\varphi)}(e \to \Box_1 e)\Big) \to \varphi^e \in L_1 \times L_2.$$

As a consequence of this and the results in Section 3.4, we obtain that expanding relativised products are usually decidable if one of their components is an **S5**- or **K**-like logic.

On the other hand, as we saw in Section 3.4, products of 'transitive' logics with frames of arbitrarily large finite or infinite cluster-depth are undecidable. The following result of [27] shows that expanding relativised product logics with components having transitive frames of arbitrarily large finite cluster-depths can be *decidable*:

THEOREM 51. *If* $L_1, L_2 \in \{$**GL**, **Grz**, **GL.3**, **Grz.3**$\}$ *then* $(L_1 \times L_2)^{\mathsf{EX}}$ *is decidable.*

Here we discuss the main points of the proof for the case of $(\mathbf{GL}\times\mathbf{GL})^{\mathsf{EX}}$ only. For the other cases, as well as for more general results, consult [27].

We remind the reader that $\mathsf{Fr}\,\mathbf{GL}$ consists of all the irreflexive, transitive and Noetherian frames. Recall that the *depth* $d^{\mathfrak{F}}(x)$ of a point x in an irreflexive tree $\mathfrak{F}=\langle W,R\rangle$ is defined to be the R-distance of x from the root. More precisely, the depth of the root is 0, and the depth of immediate R-successors of a point of depth n is $n+1$. If for no $n<\omega$ the point x is of depth n, then we say that x is of *infinite depth.*

The first important step in the proof is to show that $(\mathbf{GL}\times\mathbf{GL})^{\mathsf{EX}}$ has the 'expanding' version of the product fmp:

LEMMA 52. *Given some $\mathcal{ML}_2$-formula φ, if φ is satisfiable in a frame for $(\mathbf{GL}\times\mathbf{GL})^{\mathsf{EX}}$ then φ is satisfiable in an expanding relativised product frame $\mathfrak{H}$ that is a subframe of a product of two finite trees $\mathfrak{G}_1$ and $\mathfrak{G}_2$. Moreover, $\mathfrak{G}_1=\langle U_1,S_1\rangle$ can be chosen such that, for every $x\in U_1$,*

- $|\{y \mid \langle x,y\rangle \text{ in } \mathfrak{H}\}| \le \big(|sub\,\varphi|+1\big)!^{\,d^{\mathfrak{G}_1}(x)+1}$, *and*
- *x has at most $|sub\,\varphi|\cdot\big(|sub\,\varphi|+1\big)!^{\,d^{\mathfrak{G}_1}(x)+1}$ immediate S_1-successors.*

Proof. By a standard unravelling argument one can show that every rooted frame for $\mathbf{GL}$ is a p-morphic image of a Noetherian tree-like frame. Moreover, similarly to Proposition 9, one can show that $(\mathbf{GL}\times\mathbf{GL})^{\mathsf{EX}}$ is determined by expanding relativised product frames that are subframes of products of two Noetherian tree-like frames. So we may assume that our formula φ is satisfied at the root $\langle r_1,r_2\rangle$ of some model $\mathfrak{M}=\langle\mathfrak{H},\mathfrak{V}\rangle$ that is based on an expanding relativised subframe $\mathfrak{F}=\langle W,R_1',R_2'\rangle$ of the product of two (possibly infinite) Noetherian tree-like frames $\mathfrak{F}_1=\langle W_1,R_1\rangle$ and $\mathfrak{F}_2=\langle W_2,R_2\rangle$.

For $i=1,2$, call a point x in W_i *R_i-maximal in a set $X\subseteq W_i$*, if $x\in X$ and there is no $x'\in X$ with xR_ix'. Now we take the closure U of the set $\{\langle r_1,r_2\rangle\}$ under the the following three rules:

- **$\Diamond_1$-rule:** if $\langle x,y\rangle\in U$, $(\mathfrak{M},\langle x,y\rangle)\models\Diamond_1\psi$, for some $\Diamond_1\psi\in sub\,\varphi$, and there is no $\langle x',y\rangle\in U$ such that xR_1x' and $(\mathfrak{M},\langle x',y\rangle)\models\psi$, then choose a point $x'\in W_1$ that is R_1-maximal in the set $\{z\mid xR_1z,\ \langle z,y\rangle\in W \text{ and } (\mathfrak{M},\langle z,y\rangle)\models\psi\}$ (such a point exists because $\mathfrak{F}_1$ is Noetherian), and set $U:=U\cup\{\langle x',y\rangle\}$.
- **$\Diamond_2$-rule:** if $\langle x,y\rangle\in U$, $(\mathfrak{M},\langle x,y\rangle)\models\Diamond_2\psi$, for some $\Diamond_2\psi\in sub\,\varphi$, and there is no $\langle x,y'\rangle\in U$ such that yR_2y' and $(\mathfrak{M},\langle x,y'\rangle)\models\psi$, then choose a point $y'\in W_2$ that is R_2-maximal in the set $\{z\mid yR_2z,\ \langle x,z\rangle\in W \text{ and } (\mathfrak{M},\langle x,z\rangle)\models\psi\}$ (such a point exists because $\mathfrak{F}_2$ is Noetherian), and set $U:=U\cup\{\langle x,y'\rangle\}$.
- **Square-rule:** if $\langle x,y\rangle\in U$, xR_1x' and $\langle x',y\rangle\notin U$, then set $U:=U\cup\{\langle x',y\rangle\}$.

Now let $S_i'=R_i'\cap(U\times U)$ $(i=1,2)$ and $\mathfrak{H}=\langle U,S_1',S_2'\rangle$. Take also $\mathfrak{G}_1=\langle U_1,S_1\rangle$ and $\mathfrak{G}_2=\langle U_2,S_2\rangle$, where $U_1=\{x\in W_1\mid \langle x,r_2\rangle\in U\}$, $U_2=\{y\in W_2\mid \exists x\in U_1\ \langle x,y\rangle\in U\}$, and $S_i=R_i\cap(U_i\times U_i)$ $(i=1,2)$. Then clearly, $\mathfrak{H}$ is an expanding relativised subframe of the product of Noetherian tree-like frames $\mathfrak{G}_1$ and $\mathfrak{G}_2$.

We show that $\mathfrak{G}_1$ and $\mathfrak{G}_2$ are in fact finite trees with the required bounds. First, we claim that

$$\text{if } x \text{ is of finite depth in } \mathfrak{G}_1, \text{ then } |\{y\mid\langle x,y\rangle\in U\}| \le \big(|sub\,\varphi|+1\big)!^{\,d^{\mathfrak{G}_1}(x)+1}. \tag{24}$$

Indeed, we can proceed by induction on n. If $n = 0$, then by applying the $\Diamond_2$-rule to the root $\langle r_1, r_2\rangle$ of $\mathfrak{H}$, we can obtain $\leq |sub\,\varphi|$ immediate S_2'-successors of the form $\langle r_1, y\rangle$. In view of maximality, at each of these points the number of formulas of the form $\Diamond_2\psi \in sub\,\varphi$ to which the $\Diamond_2$-rule still applies is $\leq |sub\,\varphi| - 1$. We proceed with the same kind of argument and finally get

$$|\{y \mid \langle x, y\rangle \in U\}| \;\;\leq\;\; 1+|sub\,\varphi|+|sub\,\varphi|\cdot(|sub\,\varphi|-1)+\cdots+|sub\,\varphi|\,! \;\leq\; (|sub\,\varphi|+1)\,!\,.$$

The induction step for y of depth $n+1$ is considered analogously. The only difference is that instead of one 'starting' point we should start applying the $\Diamond_2$-rule to all points of the form $\langle x, y\rangle$ such that $\langle z, y\rangle \in U$ for the unique point z with $d(z) = n$ and zS_1y, that is to $|\{y \mid \langle z, y\rangle \in U\}| \leq \big(|sub\,\varphi| + 1\big)!^{\,n+1}$ many points.

Next, we claim that every point x of finite depth in $\mathfrak{G}_1$ has $\leq |sub\,\varphi|\cdot\big(|sub\,\varphi|+1\big)!^{\,d(x)+1}$ immediate S_1-successors. Indeed, it follows from (24) and the fact that the $\Diamond_1$-rule can be applied at most $|sub\,\varphi|$ times to a point $\langle x, y\rangle$.

Finally, we claim that every point in $\mathfrak{G}_1$ is of finite depth, that is, $\mathfrak{G}_1$ is a tree. Indeed, since $\mathfrak{G}_1$ is Noetherian, we cannot have infinite ascending chains of distinct points in it. Suppose $\mathfrak{G}_1$ still contains a point x of infinite depth. This means that there is an infinite descending chain $\ldots S_1x_2S_1x_1S_1x$. Let z be an S_1-maximal point of finite depth such that zS_1x. By (24), $|\{y \mid \langle z, y\rangle \in U\}|$ is finite. Therefore, we may apply the $\Diamond_1$-rule to points of the form $\langle z, y\rangle$ finitely many times only, and so there exists an immediate S_1-successor z' of z located properly between z and x. But then $d(z') = d(z)+1$, and so the depth of z' is finite, which is a contradiction.

Therefore, $\mathfrak{G}_1$ is a Noetherian tree with finite branching. Therefore, by König's lemma, it must be finite. So $\mathfrak{G}_2$ is finite as well. This completes the proof of Lemma 52. ❑

We are now in a position to prove that $(\mathbf{GL}\times\mathbf{GL})^{\mathsf{EX}}$ is decidable. It is to be noted that the 'expanding product fmp' does not give decidability automatically because (i) Lemma 52 does not provide us with an effective upper bound for the size of a model refuting a given formula, nor (ii) do we know that $(\mathbf{GL}\times\mathbf{GL})^{\mathsf{EX}}$ is finitely axiomatisable.

Instead, we will use a version of Kruskal's tree theorem [50]. Given a finite set Σ, a *labelled Σ-tree* is a pair $\mathfrak{T} = \langle\langle T, <\rangle, l\rangle$, where $\langle T, <\rangle$ is a (transitive, irreflexive) tree and l is a function from T to Σ. Given two finite labelled Σ-trees $\mathfrak{T}_i = \langle\langle T_i, <_i\rangle, l_i\rangle$, $i = 1, 2$, we say that $\mathfrak{T}_1$ is *embeddable* into $\mathfrak{T}_2$ if there exists an injective map $\iota : T_1 \to T_2$ such that, for all $u, v \in T_1$,

- $u <_1 v$ iff $\iota(u) <_2 \iota(v)$,
- $l_2(\iota(u)) = l_1(u)$.

Now Kruskal's tree theorem says that for every infinite sequence $\mathfrak{T}_1, \mathfrak{T}_2, \ldots$ of finite labelled Σ-trees, there exist $i < j < \omega$ such that $\mathfrak{T}_i$ is embeddable into $\mathfrak{T}_j$.

In order to use this theorem, we again turn our models to *quasimodels*. The quasimodels used here are similar to the $L\times\mathbf{K}$-quasimodels of Section 3.4, but they do differ from them in two important aspects: (i) quasistates are now not intransitive, but *transitive* and *irreflexive* trees; (ii) runs are not total, but only *partial* functions over the underlying frame.

To be precise, given an $\mathcal{ML}_2$-formula φ, a *quasistate* for φ is a finite labelled (transitive, irreflexive) tree $\langle\langle T, <\rangle, \boldsymbol{t}\rangle$ where the label $\boldsymbol{t}(x)$ of each $x \in T$ is a type for φ, and $\langle\langle T, <\rangle, \boldsymbol{t}\rangle$ satisfies the $\Diamond_2$-saturation condition **(qm1)** of Section 3.4.

A *basic structure* for φ is a pair $\langle\mathfrak{F},\boldsymbol{q}\rangle$ such that $\mathfrak{F}=\langle W,R\rangle$ is a finite (transitive, irreflexive) tree and $\boldsymbol{q}$ a function associating with each $w\in W$ a quasistate $\boldsymbol{q}(w)=\langle\langle T_w,<_w\rangle,\boldsymbol{t}_w\rangle$ for φ. We call such a basic structure *small* if, for all $x,y\in W$,

(sm1) $|T_x|\leq\big(|sub\,\varphi|+1\big)!^{\,d^{\mathfrak{F}}(x)+1}$,

(sm2) x has at most $|sub\,\varphi|\cdot\big(|sub\,\varphi|+1\big)!^{\,d^{\mathfrak{F}}(x)+1}$ immediate R-successors in $\mathfrak{F}$, and

(sm3) if xRy and $x\neq y$ then $\boldsymbol{q}(x)$ is not embeddable into $\boldsymbol{q}(y)$.

For every $n<\omega$, let Q_n be the set of all small basic structures $\langle\mathfrak{F},\boldsymbol{q}\rangle$ such that $\mathfrak{F}$ is a finite (transitive, irreflexive) tree of depth n.

LEMMA 53. *There is an $n<\omega$ such that $Q_n=\emptyset$, and so the set of small basic structures for φ is finite and can be constructed effectively from φ.*

Proof. Suppose otherwise. Define a relation E on the set Q of all small basic structures as follows. For $\mathfrak{Q}=\langle\mathfrak{F},\boldsymbol{q}\rangle$, $\mathfrak{Q}'=\langle\mathfrak{F}',\boldsymbol{q}'\rangle$ in Q, set $\mathfrak{Q}E\mathfrak{Q}'$ iff $\mathfrak{F}$ is an 'initial subtree' of $\mathfrak{F}'$ and $\boldsymbol{q}$ coincides with $\boldsymbol{q}'$ on the points of $\mathfrak{F}$. Clearly, for every $\mathfrak{Q}'\in Q_{n+1}$, there is some $\mathfrak{Q}\in Q_n$ such that $\mathfrak{Q}E\mathfrak{Q}'$. Therefore, by König's infinity lemma, there is an infinite E-chain $\mathfrak{Q}_0E\mathfrak{Q}_1E\dots E\mathfrak{Q}_nE\dots$ in Q such that $\mathfrak{Q}_n\in Q_n$ for $n<\omega$. Since $\mathfrak{Q}_{n+1}$ is always an extension of $\mathfrak{Q}_n$, their union $\mathfrak{Q}=\bigcup_{n<\omega}\mathfrak{Q}_n$ is also a basic structure. Let $\mathfrak{Q}=\langle\mathfrak{F},\boldsymbol{q}\rangle$ and $\mathfrak{F}=\langle W,R\rangle$. Then $\mathfrak{F}$ is an infinite tree with finite branching. By König's lemma, it must have an infinite branch $x_0Rx_1R\dots$. Then, by Kruskal's theorem, there exist $i<j<\omega$ such that $\boldsymbol{q}(x_i)$ is embeddable into $\boldsymbol{q}(x_j)$. But x_i and x_j already belonged to the underlying tree of $\mathfrak{Q}_j$, contrary to $\mathfrak{Q}_j$ being in Q_j. ❏

A *run through* a basic structure $\langle\mathfrak{F},\boldsymbol{q}\rangle$ is a partial function r from W giving for each $w\in\operatorname{dom}r$ a point $r(w)\in T_w$ such that, for all $x\in W$, if $x\in\operatorname{dom}r$ and xRy then $y\in\operatorname{dom}r$. Coherent and saturated runs are defined as in Section 3.4. Finally, we call a triple $\langle\mathfrak{F},\boldsymbol{q},\mathcal{R}\rangle$ a $(\mathbf{GL}\times\mathbf{GL})^{\mathsf{EX}}$-*quasimodel* (for φ) if $\langle\mathfrak{F},\boldsymbol{q}\rangle$ is a basic structure and $\mathfrak{R}$ is a set of coherent and saturated runs through it, satisfying the following conditions (cf. **(qm2)**–**(qm4)** of Section 3.4):

(eqm2) $\varphi\in\boldsymbol{t}_{w_0}(x_0)$, where w_0 and x_0 are the roots of $\mathfrak{F}$ and $\langle T_{w_0},<_{w_0}\rangle$, respectively;

(eqm3) for all $r,r'\in\mathcal{R}$ and for all $x,y\in\operatorname{dom}r\cap\operatorname{dom}r'$, $w_{r(x)}<_x w_{r'(x)}$ iff $w_{r(y)}<_y w_{r'(y)}$;

(eqm4) for all $x\in W$ and $w\in T_x$ there is $r\in\mathcal{R}$ such that $r(x)=w$.

We call a quasimodel *small* if the underlying basic structure is small.

LEMMA 54. *φ is satisfiable in a frame for $(\mathbf{GL}\times\mathbf{GL})^{\mathsf{EX}}$ iff there is a small $(\mathbf{GL}\times\mathbf{GL})^{\mathsf{EX}}$-quasimodel for φ.*

Proof. Turning a quasimodel to a 'real' model is easy, so let us concentrate on the opposite direction. We may assume that φ is satisfied in a model $\mathfrak{M}=\langle\mathfrak{H},\mathfrak{V}\rangle$ based on an expanding relativised subframe $\mathfrak{H}$ of a product $\mathfrak{G}_1\times\mathfrak{G}_2$ satisfying the conditions of Lemma 52. We can turn $\mathfrak{M}$ to a $(\mathbf{GL}\times\mathbf{GL})^{\mathsf{EX}}$-quasimodel $\langle\mathfrak{G}_1,\boldsymbol{q},\mathcal{R}\rangle$ as follows. Suppose that $\mathfrak{G}_i=\langle U_i,S_i\rangle$, for $i=1,2$. For every $x\in U_1$, let $\boldsymbol{q}(x)=\langle\langle T_x,<_x\rangle,\boldsymbol{t}_x\rangle$, where

$$\begin{aligned} T_x &= \{y\in U_2 \mid \langle x,y\rangle \text{ in } \mathfrak{H}\}, &\qquad <_x &= S_2\cap(T_x\times T_x),\\ \boldsymbol{t}_x(y) &= \{\psi\in sub\,\varphi \mid (\mathfrak{M},\langle x,y\rangle)\models\psi\}. \end{aligned}$$

Finally, for every $y \in U_2$ define a run r_y through $\langle \mathfrak{G}_1, \boldsymbol{q} \rangle$ by taking

$$\operatorname{dom} r_y = \{x \in U_1 \mid \langle x, y \rangle \text{ in } \mathfrak{H}\}$$

and then $r_y(x) = y$, for every $x \in \operatorname{dom} r_y$. Put $\mathcal{R} = \{r_y \mid y \in U_2\}$. It is straightforward to check that $\langle \mathfrak{G}_1, \boldsymbol{q}, \mathcal{R} \rangle$ is indeed a $(\mathbf{GL} \times \mathbf{GL})^{\mathsf{EX}}$-quasimodel for φ. Moreover, by the assumption on $\mathfrak{M}$, the basic structure $\langle \mathfrak{G}_1, \boldsymbol{q} \rangle$ is finite. To show that we can turn it to a basic structure satisfying **(sm3)**, suppose that there are $x, y \in U_1$ such that xS_1y and $\boldsymbol{q}(x)$ is embeddable into $\boldsymbol{q}(y)$ by an embedding ι. Then we replace in $\mathfrak{G}_1$ the subtree generated by x with the subtree generated by y, thus obtaining some tree $\mathfrak{G}' = \langle U', S' \rangle$. Let $\boldsymbol{q}'$ be the restriction of $\boldsymbol{q}$ to U'. We define new runs through $\langle \mathfrak{G}', \boldsymbol{q}' \rangle$ by taking, for all $r, r' \in \mathcal{R}$ such that $x \in \operatorname{dom} r$, $y \in \operatorname{dom} r'$, $\iota(r(x)) = r'(y)$, and for all $z \in U'$, $z \in \operatorname{dom} r$,

$$(r + r')(z) \;=\; \begin{cases} r(z), & \text{if } zS_1x, \\ r'(z), & \text{if } z = y \text{ or } yS_1z. \end{cases}$$

Let $\mathcal{R}'$ be the collection of these new runs together with those runs from $\mathcal{R}$ that 'start at' a point z with yS_1z. It is straightforward to check that $\langle \mathfrak{G}', \boldsymbol{q}', \mathcal{R}' \rangle$ is a $(\mathbf{GL} \times \mathbf{GL})^{\mathsf{EX}}$-quasimodel for φ. Since $\mathfrak{G}_1$ is finite, after finitely many repetitions of this procedure the underlying basic structure will satisfy **(sm3)**. To comply with the cardinality conditions **(sm1)** and **(sm2)**, we can use the construction from the proof of Lemma 52. Then, again we can get rid of the embeddable pairs as above, and so on. As at each step the underlying tree can get only smaller, we end up with a small $(\mathbf{GL} \times \mathbf{GL})^{\mathsf{EX}}$-quasimodel for φ. ❑

Now we can describe the decision algorithm for $(\mathbf{GL} \times \mathbf{GL})^{\mathsf{EX}}$ as follows. Given a formula φ, by Lemma 53, we can effectively construct the set of all small basic structures for φ. Then for each such small basic structure, we check whether it is a $(\mathbf{GL} \times \mathbf{GL})^{\mathsf{EX}}$-quasimodel for φ. By Lemma 54, this way we find a quasimodel for φ iff φ is satisfiable in a frame for $(\mathbf{GL} \times \mathbf{GL})^{\mathsf{EX}}$.

Observe that this decision procedure does not give an even primitive recursive complexity bound. In fact, using a reduction of the reachability problem for *lossy channel systems* (known to be non-primitive recursive by Schnoebelen [73]), it is shown in [27] that there is no primitive recursive decision algorithm for $(\mathbf{GL} \times \mathbf{GL})^{\mathsf{EX}}$.

Other expanding relativised products can even be more complex. The results in [46] suggest that logics like $(\mathsf{Log}\{\langle \mathbb{N}, < \rangle\} \times \mathsf{Log}\{\langle \mathbb{N}, < \rangle\})^{\mathsf{EX}}$ or $(\mathsf{Log}\{\langle \mathbb{N}, < \rangle\} \times \mathbf{S4})^{\mathsf{EX}}$ are undecidable. However, nothing is known about the decision problem of products 'expanding along' branching transitive frames with infinite ascending chains, such as $(\mathbf{K4} \times \mathbf{K4})^{\mathsf{EX}}$ or $(\mathbf{S4} \times \mathbf{S4.3})^{\mathsf{EX}}$. Note that these logics do not have the 'expanding product fmp'.

Another open direction of research is to study the decision problem for the finitely axiomatisable logics obtained by adding either only one of the commutativity axioms or the Church–Rosser axioms to decidable fusions.

5 OTHER COMBINATIONS

Of course, even within the constraints of the combination principles **(C1)**–**(C3)** formulated in the introduction, there are infinitely many ways of combining modal logics. Though much research have been done on multimodal logics, very little of it can be

regarded as systematic investigation into properties of some combination method. Moreover, the 'global analysis'—as explained in Chapter 7 of this handbook for the unimodal case—of multimodal logics is still in its infancy. (In fact, most of the investigations into fusions and products can be considered as the first detailed case studies.) The translation of [49] of multimodal logics into unimodal ones is not helpful either in the combination context, as it makes the information about the 'components' virtually disappear.

Releasing (parts of) the criteria **(C2)** and **(C3)** allows us to treat more 'multi-aspect' approaches to modal logic as combinations. The possibilities are again endless, below we discuss a rather ad hoc list of examples. Many more ideas that are relevant to combining modal logics can be found in the 'combining systems' literature, see e.g. the series 'Frontiers of Combining Systems (FroCoS)' [8, 14, 45, 2].

Interaction operators. Interaction between the components can be handled not only by adding interaction axioms, but also by enriching the language with 'dimension-connecting' connectives.

Perhaps the simplest and most natural operations of this sort are the *diagonal constants* d_{ij}. Given two natural numbers i and j with $1 \leq i, j \leq n$, the truth-relation for the constant d_{ij} in models over (subframes of) n-ary product frames is defined as follows:

$$(\mathfrak{M}, \langle u_1, \dots, u_n \rangle) \models \mathsf{d}_{ij} \quad \text{iff} \quad u_i = u_j.$$

The set of n-tuples satisfying d_{ij} is usually called the (i, j)-*diagonal element*. The main reason for introducing such constants has been to give a 'modal treatment' of equality of classical first-order logic, see Section 3.2 above. Modal algebras for the product logic $\mathbf{S5}^n$ extended with diagonal elements are called *representable cylindric algebras* and are extensively studied in the algebraic logic literature; see e.g. [39, 41] and the references therein.

Another natural way of connecting dimensions is via so-called 'jump' modalities. Given a function $\pi : \{1, \dots, n\} \to \{1, \dots, n\}$ (such a map can be called a *jump*), define the truth-relation for the unary modal operator s_π in models over (subframes of) product frames as follows:

$$(\mathfrak{M}, \langle u_1, \dots, u_n \rangle) \models \mathsf{s}_\pi \varphi \quad \text{iff} \quad (\mathfrak{M}, \langle u_{\pi(1)}, \dots, u_{\pi(n)} \rangle) \models \varphi.$$

These modal operators are often called (*generalised*) *substitutions*, since by taking

$$P(x_{\pi(1)-1}, \dots, x_{\pi(n)-1})^\bullet = \mathsf{s}_\pi P(x_0, \dots, x_{n-1}) \qquad (P \text{ an atomic formula})$$

one can extend the translation $^\bullet$ of Section 3.2 above from formulas with a fixed order of the variables to arbitrary first-order formulas. Note that in cubic universal product $\mathbf{S5}^n$-frames certain substitutions are expressible with the help of the boxes and the diagonal constants [39]. Various versions of modal algebras corresponding to products of $\mathbf{S5}$ logics with substitutions and with or without diagonal constants (e.g., *polyadic* and *substitution algebras*) are introduced by Halmos [33, 34] and Pinter [64, 65] and have been studied in the algebraic logic literature ever since.

Valuation restrictions. One may try to loosen the strong interaction between the components of product logics by imposing restrictions on possible valuations in models over (subframes of) product frames. In general, the resulting formalisms will not be closed under the rule of Substitution, and so do not satisfy **(C2)**.

DEFINITION 55. Let $\mathfrak{M}_1 = \langle \mathfrak{F}_1, \mathfrak{V}_1 \rangle$ and $\mathfrak{M}_2 = \langle \mathfrak{F}_2, \mathfrak{V}_2 \rangle$ be Kripke models that are based on respective frames $\mathfrak{F}_1 = \langle W_1, R_1 \rangle$ and $\mathfrak{F}_2 = \langle W_2, R_2 \rangle$, and let $\Phi(x_1, x_2)$ be a formula in the first-order language $\mathcal{L}$ having two unary predicate symbols V_1, V_2 and two binary predicate symbols. Then a model

$$\mathfrak{M}^\Phi = \left\langle \mathfrak{F}_1 \times \mathfrak{F}_2, \mathfrak{V}^\Phi \right\rangle$$

is said to be a Φ-*flat product of* $\mathfrak{M}_1$ *and* $\mathfrak{M}_2$ if, for all propositional variables p and all $u_1 \in W_1$, $u_2 \in W_2$,

$$\langle u_1, u_2 \rangle \in \mathfrak{V}^\Phi(p) \quad \text{iff} \quad I_p \models \Phi[u_1, u_2],$$

where I_p is the first-order $\mathcal{L}$-structure $I_p = \langle W_1 \cup W_2, \mathfrak{V}_1(p), \mathfrak{V}_2(p), R_1, R_2 \rangle$. The valuations in flat-product models are called *flat valuations.*

If Φ is a Boolean combination of $V_1(x_1)$ and $V_2(x_2)$ then we say that $\mathfrak{M}^\Phi$ is a *Boolean-flat model.*

Boolean-flat models are introduced and studied in [21, 37]. Various special cases of flat valuations are discussed for many-dimensional temporal logics in [21, 22] and for temporal arrow logics in [61]. Satisfiability in Boolean-flat models can be reduced to satisfiability in the component models, as the following 'flat product decomposition theorem' of Gabbay and Shehtman [25, 23] shows:

THEOREM 56. *Let $\mathfrak{M}^\Phi$ be a Boolean-flat product of models $\mathfrak{M}_1$ and $\mathfrak{M}_2$. Then for every $\mathcal{ML}_2$-formula φ, there are a finite set I_φ and unimodal formulas $\varphi_i^{\mathbf{1}}$ (with $\Box_1$) and $\varphi_i^{\mathbf{2}}$ (with $\Box_2$), $i \in I_\varphi$, such that, for all worlds $\langle u_1, u_2 \rangle$ in $\mathfrak{M}^\Phi$,*

$$(\mathfrak{M}^\Phi, \langle u_1, u_2 \rangle) \models \varphi \qquad \textit{iff} \qquad \exists i \in I_\varphi \big((\mathfrak{M}_1, u_1) \models \varphi_i^{\mathbf{1}} \textit{ and } (\mathfrak{M}_2, u_2) \models \varphi_i^{\mathbf{2}} \big).$$

Modalising one logic with another. Another possibility is to take some combination method satisfying **(C1)**–**(C3)** and then consider a *fragment* of the full modal language only. The general methodology of 'temporalising' a logic, introduced by Finger and Gabbay [17], results in such a combination when applied to two modal logics:

DEFINITION 57. The set of *modalised formulas* is the smallest set Γ of $\mathcal{ML}_2$-formulas such that:

- if φ is an $\mathcal{ML}_1$-formula then $\varphi \in \Gamma$,
- Γ is closed under Boolean combinations,
- if $\varphi \in \Gamma$ then $\Diamond_2 \varphi \in \Gamma$ and $\Box_2 \varphi \in \Gamma$.

We will evaluate modalised formulas in *modalised models.* These are structures of the form $\mathfrak{M} = \langle \mathfrak{F}, f \rangle$, where $\mathfrak{F} = \langle W, R \rangle$ is a frame and f is a function mapping each $w \in W$ to a pair $f(w) = \langle \mathfrak{M}_w, x_w \rangle$ with $\mathfrak{M}_w$ being a Kripke model and x_w a world in it. The *truth-relation* '$\mathfrak{M}, w \models \varphi$' for modalised formulas φ and worlds w in $\mathfrak{F}$ is defined inductively as follows:

- $\mathfrak{M}, w \models \psi$ iff $(\mathfrak{M}_w, x_w) \models \psi$, whenever ψ is an $\mathcal{ML}_1$-formula,
- $\mathfrak{M}, w \models \neg\psi$ iff $\mathfrak{M}, w \not\models \psi$,
- $\mathfrak{M}, w \models \psi \wedge \chi$ iff $\mathfrak{M}, w \models \psi$ and $\mathfrak{M}, w \models \chi$,

- $\mathfrak{M}, w \models \Diamond_2\psi$ iff there is $v \in W$ such that wRv and $\mathfrak{M}, v \models \psi$.

We say that φ *is true in* $\mathfrak{M}$ if $\mathfrak{M}, w \models \varphi$ for all $w \in W$.

Now let L_1 and L_2 be two Kripke complete unimodal logics formulated in the language $\mathcal{ML}_1$ in such a way that they have different modal operators (say, $\Diamond_1, \Box_1$ and $\Diamond_2, \Box_2$, respectively). The *modalisation of* L_1 *with* L_2 is the set $L_2[L_1]$ of modalised formulas that are true in all modalised models $\mathfrak{M} = \langle \mathfrak{F}, f \rangle$ where

- $\mathfrak{F}$ is a frame for L_2, and
- for all w in $\mathfrak{F}$, the underlying frame of $\mathfrak{M}_w$ is a frame for L_1.

It is not hard to see that $L_2[L_1]$ is a decidable subset of all $\mathcal{ML}_2$-formulas, whenever L_1 and L_2 are both decidable. In fact, this is a consequence of Theorem 5, as $L_2[L_1]$ is a fragment of the fusion of L_1 and L_2 in the sense that $L_2[L_1]$ is the set of all modalised formulas in $L_1 \otimes L_2$ (cf. the proof of Theorem 3).

$\mathcal{E}$-connections. This combination method is introduced by Kutz *et al.* [56, 55] in the more general setting of 'abstract description systems.' When applied to modal logics, this method takes disjoint Kripke models for each component and connects their domains via 'link relations.' These 'connections' then also appear explicitly in the language.

DEFINITION 58. Suppose that we have n 'copies' of the unimodal language $\mathcal{ML}_1$ in such a way that their sets of propositional variables are disjoint (say, $p_0^i, p_1^i, \dots$ for the ith copy) and their modal operators are disjoint as well (say, $\Box_i$ and $\Diamond_i$ for the ith copy). Let J be a non-empty set and take an $n-1$-ary new connective ${}^i\langle E_j \rangle$, for each $j \in J$, $i = 1, \dots, n$. Then the *n-ary $\mathcal{E}$-connection language* $\mathcal{EL}_n^J$ is defined as follows. *$\mathcal{EL}_n^J$-formulas* are partitioned into n sets, each one containing the so-called *i-formulas* for some $i = 1, \dots, n$. For all $i = 1, \dots, n$, the sets of i-formulas are defined by simultaneous induction:

- the propositional variables $p_0^i, p_1^i, \dots$ are i-formulas,
- the set of i-formulas is closed under the Boolean connectives and the modal operators $\Box_i$ and $\Diamond_i$,
- if φ_k is a k-formula, for each $k = 1, \dots, i-1, i+1, \dots, n$, then
$$ {}^i\langle E_j \rangle (\varphi_1, \dots, \varphi_{i-1}, \varphi_{i+1}, \dots, \varphi_n) $$
is an i-formula, for every $j \in J$.

$\mathcal{EL}_n^J$-formulas are evaluated in *$\mathcal{EL}_n^J$-models.* These are structures of the form
$$ \mathfrak{M} = \left\langle \mathfrak{M}_1, \dots, \mathfrak{M}_n, E_j^{\mathfrak{M}} \right\rangle_{j \in J}, $$
where the $\mathfrak{M}_i = \langle \langle W_i, R_i \rangle, \mathfrak{V}_i \rangle$ are (unimodal) Kripke models and $E_j^{\mathfrak{M}} \subseteq W_1 \times \dots \times W_n$, for each $j \in J$. The *truth-relation* '$\mathfrak{M}, w \models \varphi$' for i-formulas φ and worlds w in $\mathfrak{M}_i$ is defined inductively as follows, simultaneously for $i = 1, \dots, n$:

- $\mathfrak{M}, w \models p_k^i$ iff $w \in \mathfrak{V}_i(p_k^i)$,
- $\mathfrak{M}, w \models \neg\psi$ iff $\mathfrak{M}, w \not\models \psi$,

- $\mathfrak{M}, w \models \psi \wedge \chi$ iff $\mathfrak{M}, w \models \psi$ and $\mathfrak{M}, w \models \chi$,
- $\mathfrak{M}, w \models \Diamond_i \psi$ iff there is $v \in W_i$ such that wR_iv and $\mathfrak{M}, v \models \psi$,
- $\mathfrak{M}, w \models {}^i\langle E_j \rangle (\psi_1, \ldots, \psi_{i-1}, \psi_{i+1}, \ldots, \psi_n)$ iff for all $k = 1, \ldots, i-1, i+1, \ldots, n$ there are $v_k \in W_k$ such that $\mathfrak{M}, v_k \models \psi_k$ and $\langle v_1, \ldots, v_{i-1}, w, v_{i+1}, \ldots, v_n \rangle \in E_j^{\mathfrak{M}}$.

We say that an i-formula φ *is true in* $\mathfrak{M}$ if $\mathfrak{M}, w \models \varphi$ for all $w \in W_i$.

Now let $L_1, \ldots, L_n$ be unimodal logics formulated in n 'almost disjoint' copies of $\mathcal{ML}_1$ as described above, and let J be a non-empty set. The *basic $\mathcal{E}$-connection*

$$\mathcal{EC}^J(L_1, \ldots, L_n)$$

of $L_1, \ldots, L_n$ is the set of all *$\mathcal{EL}_n^J$-formulas* that are true in all $\mathcal{EL}_n^J$-models $\mathfrak{M} = \langle \mathfrak{M}_1, \ldots, \mathfrak{M}_n, E_j^{\mathfrak{M}} \rangle_{j \in J}$ where $\mathfrak{M}_i$ is a model for L_i, for $i = 1, \ldots, n$.

The following theorem on the transfer of decidability is proved in [55] in the more general setting of 'abstract desription systems:'

THEOREM 59. *If $L_1, \ldots, L_n$ are all decidable unimodal modal logics, then the basic $\mathcal{E}$-connection $\mathcal{EC}^J(L_1, \ldots, L_n)$ is decidable.*

Intuively, the decision procedure for, say, $\mathcal{EC}^{\{0\}}(L_1, L_2)$ works as follows. As usual, we can consider satisfiability instead of validity. To check whether there exists a model $\mathfrak{M} = \langle \mathfrak{M}_1, \mathfrak{M}_2, E_0 \rangle$ and a world w in $\mathfrak{M}_i$ such that $\mathfrak{M}, w \models \varphi$ for a given i-formula φ, the algorithm non-deterministically 'guesses'

- the 1-types that are realised in $\mathfrak{M}_1$ and the 2-types that are realised in $\mathfrak{M}_2$ (where a *k-type* is a set of k-formulas that are true at a world of $\mathfrak{M}_k$), and
- a binary relation e between the guessed sets of 1-types and 2-types.

Then it checks whether the guessed sets satisfy a collection of 'integrity conditions.' This check involves satisfiability tests of certain sets of k-formulas constructed from φ, for $k = 1, 2$—here we exploit that L_1 and L_2 are decidable. If the integrity conditions are satisfied, then it is possible to construct a model satisfying φ using models of the constructed sets of k-formulas. If the integrity conditions are not satisfied then φ is not satisfiable. This algorithm also provides an upper bound for the satisfiability problem for $\mathcal{EC}^J(L_1, \ldots, L_n)$: the time complexity is non-deterministic and one exponent higher than the maximal time complexity of the component procedures.

ACKNOWLEDGEMENTS

The work on this chapter was partially supported by U.K. EPSRC grants GR/S63175/01 and GR/S61973/01. I am grateful to Carsten Lutz for his thorough reading of the manuscript and for his constructive comments and suggestions. Thanks are also due to Frank Wolter for his suggestions.

BIBLIOGRAPHY

[1] H. Andréka, I. Németi, and I. Sain. Algebraic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, Volume 2, 2nd edition*, pages 133–247. Kluwer Academic Publishers, 2001.

[2] A. Armando, editor. *Frontiers of Combining Systems IV*, volume 2309 of *Lecture Notes in Artificial Intelligence*. Springer, 2002.
[3] A. Artale and E. Franconi. A temporal description logic for reasoning about actions and plans. *Journal of Artificial Intelligence Research*, 9:463–506, 1998.
[4] F. Baader, S. Ghilardi, and C. Tinelli. A new combination procedure for the word problem that generalizes fusion decidability results in modal logic. In *Proceedings of IJCAR-04*, volume 3097 of *Lecture Notes in Artificial Intelligence*, pages 183–197. Springer, 2004.
[5] F. Baader and A. Laux. Terminological logics with modal operators. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 808–814. Morgan Kaufmann, 1995.
[6] F. Baader, C. Lutz, H. Sturm, and F. Wolter. Fusions of description logics and abstract description systems. *Journal of Artificial Intelligence Research*, 16:1–58, 2002.
[7] F. Baader and H.J. Ohlbach. A multi-dimensional terminological knowledge representation language. *Journal of Applied Non-Classical Logics*, 5:153–197, 1995.
[8] F. Baader and K. Schulz, editors. *Frontiers of Combining Systems*. Kluwer Academic Publishers, 1996.
[9] R. Berger. The undecidability of the domino problem. *Memoirs of the AMS*, 66, 1966.
[10] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
[11] W. Blok and D. Pigozzi. Algebraizable Logics. *Memoirs of the AMS*, 77(396), 1989.
[12] A. Chagrov and M. Zakharyaschev. *Modal Logic*, volume 35 of *Oxford Logic Guides*. Clarendon Press, Oxford, 1997.
[13] W. Craig. *Logic in Algebraic Form*. North-Holland, 1974.
[14] M. de Rijke and D. Gabbay, editors. *Frontiers of Combining Systems II*. Research Studies Press LTD., 2000.
[15] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.
[16] K. Fine and G. Schurz. Transfer theorems for stratified modal logics. In J. Copeland, editor, *Logic and Reality, Essays in Pure and Applied Logic. In memory of Arthur Prior*, pages 169–213. Oxford University Press, 1996.
[17] M. Finger and D. Gabbay. Adding a temporal dimension to a logic system. *Journal of Logic, Language and Information*, 2:203–233, 1992.
[18] M. Finger and M. Reynolds. Imperative history: two-dimensional executable temporal logic. In U. Reyle and H.J. Ohlbach, editors, *Logic, Language and Reasoning: Essays in Honour of Dov Gabbay*, pages 81–107. Kluwer Academic Publishers, 1999.
[19] D. Gabbay. Fibred semantics and the weaving of logics, part 1: Modal and intuitionistic logics. *Journal of Symbolic Logic*, 61:1057–1120, 1996.
[20] D. Gabbay. *Fibring Logics*, volume 38 of *Oxford Logic Guides*. Clarendon Press, 1999.
[21] D. Gabbay and F. Guenthner. A note on systems of n-dimensional tense logics. In T. Pauli, editor, *Essays dedicated to L. Aqvist*, pages 63–71. Uppsala, 1982.
[22] D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects, Volume 1*. Oxford University Press, 1994.
[23] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*, volume 148 of *Studies in Logic*. Elsevier, 2003.
[24] D. Gabbay and V. Shehtman. Products of modal logics. Part I. *Journal of the IGPL*, 6:73–146, 1998.
[25] D. Gabbay and V. Shehtman. Flow products of modal logics. Manuscript, 1999.
[26] D. Gabelaia, A. Kurucz, F. Wolter, and M. Zakharyaschev. Products of 'transitive' modal logics. *Journal of Symbolic Logic*, 70:993–1021, 2005.
[27] D. Gabelaia, A. Kurucz, F. Wolter, and M. Zakharyaschev. Non-primitive recursive decidability of products of modal logics with expanding domains. *Annals of Pure and Applied Logic*, to appear.
[28] D. Gabelaia, A. Kurucz, and M. Zakharyaschev. Products of transitive modal logics without the (abstract) finite model property. In R. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing, editors, *Proceedings of AiML 2004, September 2004, Manchester, U.K.*, 2004.
[29] S. Ghilardi. Quantifier elimination and provers integration. In *Proceedings of 'First-order Theorem Proving 2003'*, volume 86 of *Electronic Notes in Theoretical Computer Science*, 2003.
[30] S. Ghilardi and L. Santocanale. Algebraic and model theoretic techniques for fusion decidability in modal logics. In M. Vardi and A. Voronkov, editors, *Proceedings of LPAR-03*, volume 2850 of *Lecture Notes in Artificial Intelligence*, pages 152–166. Springer, 2003.
[31] R. Goldblatt. *Logics of Time and Computation*. Number 7 in CSLI Lecture Notes, Stanford. CSLI, 1987.
[32] A. Gräber, H. Bürckert, and A. Laux. Terminological reasoning with knowledge and belief. In A. Laux and H. Wansing, editors, *Knowledge and Belief in Philosophy and Artificial Intelligence*, pages 29–61. Akademie Verlag, 1995.
[33] P. Halmos. Algebraic logic, IV. *Transactions of the American Mathematical Society*, 86:1–27, 1957.
[34] P. Halmos. *Algebraic Logic*. Chelsea Publishing Company, New York, 1962.

[35] J. Halpern and Yo. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.

[36] D. Harel. Recurring dominoes: Making the highly undecidable highly understandable. In *Conference on Foundations of Computing Theory*, volume 158 of *Lecture Notes in Computer Science*, pages 177–194. Springer, 1983.

[37] Y. Hasimoto. Normal products of modal logics. In F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev, editors, *Advances in Modal Logic, Volume 3*, pages 241–255. World Scientific, 2002.

[38] L. Henkin, J. Monk, and A. Tarski. *Cylindric Algebras, Part I*, volume 65 of *Studies in Logic*. North-Holland, 1971.

[39] L. Henkin, J. Monk, and A. Tarski. *Cylindric Algebras, Part II*, volume 115 of *Studies in Logic*. North-Holland, 1985.

[40] R. Hirsch and I. Hodkinson. Representability is not decidable for finite relation algebras. *Transactions of the American Mathematical Society*, 353:1403–1425, 2001.

[41] R. Hirsch and I. Hodkinson. *Relation Algebras by Games*, volume 147 of *Studies in Logic*. Elsevier, North-Holland, 2002.

[42] R. Hirsch, I. Hodkinson, and A. Kurucz. On modal logics between $\mathbf{K} \times \mathbf{K} \times \mathbf{K}$ and $\mathbf{S5} \times \mathbf{S5} \times \mathbf{S5}$. *Journal of Symbolic Logic*, 67:221–234, 2002.

[43] I. Hodkinson, F. Wolter, and M. Zakharyaschev. Decidable fragments of first-order temporal logics. *Annals of Pure and Applied Logic*, 106:85–134, 2000.

[44] J.S. Johnson. Nonfinitizability of classes of representable polyadic algebras. *Journal of Symbolic Logic*, 34:344–352, 1969.

[45] H. Kirchner and C. Ringeissen, editors. *Frontiers of Combining Systems III*, volume 1794 of *Lecture Notes in Artificial Intelligence*. Springer, 2000.

[46] B. Konev, F. Wolter, and M. Zakharyaschev. Temporal logics over transitive states. In R. Nieuwenhuis, editor, *Proceedings of CADE-05*, volume 3632 of *Lecture Notes in Computer Science*, pages 182–203. Springer, 2005.

[47] R. Kontchakov, A. Kurucz, and M. Zakharyaschev. Undecidability of first-order intuitionistic and modal logics with two variables. *Bulletin of Symbolic Logic*, 11:428–438, 2005.

[48] M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal logics. *Journal of Symbolic Logic*, 56:1469–1485, 1991.

[49] M. Kracht and F. Wolter. Normal monomodal logics can simulate all others. *Journal of Symbolic Logic*, 64:99–138, 1999.

[50] J. B. Kruskal. Well-quasi-ordering, the tree theorem and Vázsonyi's conjecture. *Transactions of the American Mathematical Society*, 95:210–225, 1960.

[51] S.T. Kuhn. Quantifiers as modal operators. *Studia Logica*, 39:145–158, 1980.

[52] A. Kurucz. On axiomatising products of Kripke frames. *Journal of Symbolic Logic*, 65:923–945, 2000.

[53] A. Kurucz. $\mathbf{S5} \times \mathbf{S5} \times \mathbf{S5}$ lacks the finite model property. In F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev, editors, *Advances in Modal Logic, Volume 3*, pages 321–328. World Scientific, 2002.

[54] A. Kurucz and M. Zakharyaschev. A note on relativised products of modal logics. In P. Balbiani, N.-Y. Suzuki, F. Wolter, and M. Zakharyaschev, editors, *Advances in Modal Logic, Volume 4*, pages 221–242. King's College Publications, 2003.

[55] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. $\mathcal{E}$-connections of abstract description systems. *Artificial Intelligence*, 156:1–73, 2004.

[56] O. Kutz, F. Wolter, and M. Zakharyaschev. Connecting abstract description systems. In D. Fensel, F. Giunchiglia, D. McGuinness, and M.-A. Williams, editors, *Proceedings of the 8th International Conference on Principles and Knowledge Representation and Reasoning (KR-02), Toulouse, France, 2002*, pages 215–226. Morgan Kaufmann, 2002.

[57] A. Laux. Beliefs in multi-agent worlds: a terminological approach. In A. Cohn, editor, *Proceedings of the 11th European Conference on Artificial Intelligence (ECAI-94)*, pages 299–303. John Wiley and Sons, 1994.

[58] R. Maddux. The equational theory of CA_3 is undecidable. *Journal of Symbolic Logic*, 45:311–315, 1980.

[59] M. Marx. Complexity of products of modal logics. *Journal of Logic and Computation*, 9:197–214, 1999.

[60] M. Marx and M. Reynolds. Undecidability of compass logic. *Journal of Logic and Computation*, 9:897–914, 1999.

[61] M. Marx and Y. Venema. *Multi-Dimensional Modal Logic*. Kluwer Academic Publishers, 1997.

[62] I. Németi. Algebraization of quantifier logics: an overview. *Studia Logica*, 3,4:485–569, 1991.

[63] I. Németi. Decidable versions of first-order predicate logic and cylindric relativized set algebras. In L. Csirmaz, D. Gabbay, and M. de Rijke, editors, *Logic Colloquium'92*, pages 177–241. CSLI Publications, 1995.

[64] C. Pinter. A simple algebra of first order logic. *Notre Dame Journal of Formal Logic*, 1:361–366, 1973.

[65] C. Pinter. Algebraic logic with generalized quantifiers. *Notre Dame Journal of Formal Logic*, 16:511–516, 1975.

[66] W.V. Quine. Algebraic logic and predicate functors. In R. Rudner and I. Scheffer, editors, *Logic and Art: Essays in Honor of Nelson Goodman*. Bobbs-Merrill, 1971. Reprinted with amendments in *The Ways of Paradox and Other Essays*, 2nd edition, Harvard University Press, Cambridge, Massachussetts, 1976.

[67] M.O. Rabin. Decidability of second order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

[68] J. Reif and A. Sistla. A multiprocess network logic with temporal and spatial modalities. *Journal of Computer and System Sciences*, 30:41–53, 1985.

[69] M. Reynolds. A decidable temporal logic of parallelism. *Notre Dame Journal of Formal Logic*, 38(3):419–436, Summer 1997.

[70] M. Reynolds and M. Zakharyaschev. On the products of linear modal logics. *Journal of Logic and Computation*, 11:909–931, 2001.

[71] K. Schild. Combining terminological logics with tense logic. In *Proceedings of the 6th Portuguese Conference on Artificial Intelligence*, pages 105–120, Porto, 1993.

[72] A. Schmiedel. A temporal terminological logic. In *Proceedings of the 9th National Conference of the American Association for Artificial Intelligence*, pages 640–645, Boston, 1990.

[73] Ph. Schnoebelen. Verifying lossy channel systems has nonprimitive recursive complexity. *Information Processing Letters*, 83:251–261, 2002.

[74] K. Segerberg. Two-dimensional modal logic. *Journal of Philosophical Logic*, 2:77–96, 1973.

[75] V. Shehtman. Two-dimensional modal logics. *Mathematical Notices of the USSR Academy of Sciences*, 23:417–424, 1978. (Translated from Russian).

[76] V. Shehtman. Filtration via bisimulation. In R. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing, editors, *Advances in Modal Logic, Volume 5*, pages 289–308. King's College Publications, 2005.

[77] E. Spaan. *Complexity of Modal Logics*. PhD thesis, Department of Mathematics and Computer Science, University of Amsterdam, 1993.

[78] S. Thomason. Independent propositional modal logics. *Studia Logica*, 39:143–144, 1980.

[79] J. van Benthem, G. Bezhanishvili, B. ten Cate, and D. Sarenac. Multimodal logics of products of topologies. *Studia Logica*, to appear.

[80] Y. Venema. *Many-Dimensional Modal Logics*. PhD thesis, Universiteit van Amsterdam, 1991.

[81] H. Wang. Dominos and the $\forall\exists\forall$ case of the decision problem. In *Mathematical Theory of Automata*, pages 23–55. Polytechnic Institute, Brooklyn, 1963.

[82] F. Wolter. Fusions of modal logics revisited. In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic, Volume 1*, pages 361–379. CSLI Publications, Stanford, 1998.

[83] F. Wolter. The product of converse **PDL** and polymodal **K**. *Journal of Logic and Computation*, 10:223–251, 2000.

[84] F. Wolter and M. Zakharyaschev. Satisfiability problem in description logics with modal operators. In A. Cohn, L. Schubert, and S. Shapiro, editors, *Proceedings of the 6th Conference on Principles of Knowledge Representation and Reasoning (KR-98), Trento, Italy*, pages 512–523. Morgan Kaufmann, 1998.

[85] F. Wolter and M. Zakharyaschev. Modal description logics: modalizing roles. *Fundamenta Informaticae*, 39:411–438, 1999.

[86] F. Wolter and M. Zakharyaschev. Temporalizing description logics. In D. Gabbay and M. de Rijke, editors, *Frontiers of Combining Systems II*, pages 379–401. Research Studies Press LTD., 2000.

[87] F. Wolter and M. Zakharyaschev. Decidable fragments of first-order modal logics. *Journal of Symbolic Logic*, 66:1415–1438, 2001.

[88] F. Wolter and M. Zakharyaschev. Dynamic description logic. In M. Zakharyaschev, K. Segerberg, M. de Rijke, and H. Wansing, editors, *Advances in Modal Logic, Volume 2*, pages 431–445. CSLI Publications, Stanford, 2001.

[89] F. Wolter and M. Zakharyaschev. Axiomatizing the monodic fragment of first-order temporal logic. *Annals of Pure and Applied Logic*, 118:133–145, 2002.

[90] M. Zakharyaschev, F. Wolter, and A. Chagrov. Advanced modal logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, Volume 3, 2nd edition*, pages 83–266. Kluwer Academic Publishers, 2001.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

16

MODAL LOGIC IN MATHEMATICS

Sergei Artemov

1 Introduction . . . 928
2 Some history . . . 929
3 Two models of provability . . . 930
4 Provability Logic . . . 933
4.1 Solovay's completeness theorem . . . 934
4.2 Fixed point theorem . . . 935
4.3 First-order provability logics . . . 936
4.4 Intuitionistic provability logic . . . 936
4.5 Classification of provability logics . . . 937
4.6 Provability logics with additional operators . . . 938
4.7 Generalized provability predicates . . . 938
4.8 Interpretability and conservativity logics . . . 939
4.9 Magari algebras and propositional second-order provability logic . . . 940
4.10 'True and Provable' modality . . . 941
4.11 Applications . . . 941
5 Logic of Proofs . . . 942
5.1 Arithmetical completeness . . . 944
5.2 Realization Theorem . . . 945
5.3 Fitting Models . . . 947
5.4 Joint logics of proofs and provability . . . 948
5.5 Quantified logics of proofs . . . 949
5.6 Intuitionistic logic of proofs . . . 949
5.7 Applications . . . 950
6 Modal logic of space . . . 951
6.1 Other operators in topological spaces . . . 953
6.2 Adding the universal modality . . . 954
6.3 Modal logic of metric spaces . . . 954
6.4 Modal logic of dynamic topology . . . 955
6.5 Other geometric notions . . . 958
6.6 Modal logic of spacetime . . . 958
6.7 Topoi . . . 958
6.8 Universal algebra . . . 958
7 Modalities in Set Theory . . . 959

1 INTRODUCTION

Formal modal logic is mostly mathematical in its methods, regardless of area of application. This Handbook presents a wide variety of mathematical techniques developed over decades of studying the intricate details of modal logic. Also included among relatively recent general purpose sources on the mathematics of modal logic are monographs [57, 75, 99, 114, 153] and a survey paper [115]. For that matter, the applications of mathematics in modal logic are overwhelming, while those in the dual category, the uses of modal logic in mathematics, are less numerous.

Mathematics normally finds a proper language and level of abstraction for the study of its objects. Propositional modal logic offers a new paradigm of applying logical methods: instead of using the traditional languages with quantification (first-order or higher-order) to describe a structure, look for an appropriate quantifier-free language with additional logic operators (modalities) that represent the phenomenon at hand. In a number of prominent cases, we end up with a logic-based language which is much richer than Boolean logic, and yet, unlike universal languages with quantification, does not fall under the scope of classical undecidability limitations. Modal logic often offers better decidability and complexity results than the rival first-order logic.

We adopt a strict approach as to what constitutes an application of modal logic in mathematics, i.e., we limit our attention to mathematical objects which existed independently of modal logic, rather than those developed for the needs of modal logic itself. This requirement is not by any means sufficient; after all, any class of binary relations in mathematics specifies some propositional modal logic which, however, does not automatically make these connections worthy of study. We consider only the cases in which a mathematical modality-like notion was developed and studied by mathematicians to the extent that the modal logical language and methods became pertinent. Neither is this requirement necessary; for example, elaborate algebraic models originally developed for the needs of logic (e.g., modal logic) are now deeply embedded into the corresponding field of mathematics and may well be regarded as a contribution of modal logic to mathematics. Fortunately, algebraic models for modal logic have been covered in Chapter 6 of this Handbook. Moreover, the present author has not been quite pedantic in carrying out even this imperfect approach; such important issues as topos models and the connection between modal logic and Grothendieck topology on categories were barely mentioned in this survey. Some of these topics were considered in Chapter 9 of this Handbook.

There are two major ideas that dominate the landscape of modal logic application in mathematics: Gödel's provability semantics and Tarski's topological semantics.

Gödel's use of modal logic to describe provability in the 1930s gave the first exact semantics of modality. This approach led to a comprehensive provability semantics for a broad class of modal logics, including the major ones: K, T, $\mathsf{K4}$, $\mathsf{S4}$, $\mathsf{S5}$, GL, Grz, and others. It also proved vital for such applications as the Brouwer-Heyting-Kolmogorov (intended) provability semantics for intuitionistic logic, for introducing justification into formal epistemology and tackling its logical omniscience problem, for introducing self-reference into combinatory logic and lambda-calculi, etc.

Another major use of modal logic in mathematics is the topological semantics suggested by Tarski and developed by Tarski and McKinsey in the 1940s. Here modal logic provides a natural high-level language for describing topology in a point-free manner. In addition to its natural mathematical appeal, this approach has evolved into an active

research area with applications in dynamic systems, control systems, spatio-temporal reasoning, etc.

There has also been significant research activity in applying modal logic to set theory, which can be traced back to Solovay's work of the 1970s. We devote Section 7 to this issue.

The reader might perceive a certain bias towards provability logic in this survey. A possible explanation is that Gödel's provability semantics of modal logic is the oldest and arguably the most well-established tradition of applying modal logic to mathematics. It is perhaps more essential for proof theory and foundations than other applications of modal logic for the corresponding object areas of mathematics. This observation is not intended to discount other interpretations of modal logic considered here; we hope that this survey gives a fair assessment of their beauty and vast potential.

Among other recent surveys in this area, we recommend the article 'Provability logic' by Verbrugge in the Stanford Encyclopedia of Philosophy

`http://plato.stanford.edu/entries/logic-provability/`,

the handbook chapter 'Provability Logic' [25], and the forthcoming collection 'The Logic of Space' edited by Aiello, van Benthem, and Pratt-Hartmann.

2 SOME HISTORY

In his 1933 paper [109], Gödel chose the language of propositional modal logic to describe the basic logical laws of provability. According to his approach, $\Box F$ should be interpreted informally as

$$F \textit{ is provable},$$

and the classical modal logic $\mathsf{S4}$ provides a system of plausible postulates for provability. Gödel's goal was to provide an exact interpretation of intuitionistic propositional logic within a classical modal logic of provability, hence giving classical meaning to the basic intuitionistic logical system.

This line of research attracted a great deal of attention in mathematics and eventually led to two distinct models of provability based on modal logics:

1. the Provability Logic GL, which was shown by Solovay to be the logic of Gödel's formal provability predicate;

2. Gödel's original logic $\mathsf{S4}$, which was shown by Artemov to be a forgetful projection of the Logic of Proofs LP.

These two models complement each other and cover a wide range of applications, from traditional proof theory to formal verification and epistemology.

The use of modal logic in topology was initially motivated by Kuratowski's axioms for topological spaces, which were recast in the manner of modal logic by Tarski in the late 1930s. Under this interpretation, the Boolean components were treated in the usual set theoretical way as subsets of a given topological set, whereas $\Box$ was interpreted as

the interior operator.

In their seminal paper of 1944 [187], McKinsey and Tarski proved that S4 was the logic of any separable dense-in-itself metric space, in particular the real topological space $\mathbb{R}^n$, for each $n = 1, 2, 3, \ldots$. Among other known topological operators on sets,

the derived set operator,

regarded as the modality $\Diamond$, has been given a complete axiomatization in works by Esakia and Shehtman. The modal logic of topology developed into an area of research that included modal studies of other operators in topological spaces, modal logic of metric spaces, dynamic topological logic, spatio-temporal reasoning, etc., with applications outside the original mathematical core.

It was perhaps Solovay who initiated research in the application of modal logic to set theory in 1976 when he gave a complete axiomatization of such modalities as

true in all transitive models of ZF

and

true in all universes.

Hamkins and Löwe recently found a complete axiom system of the modality

true in all forcing extensions.

Studies of connections between infinitary modal logic and set theory initiated by Barwise and Moss in 1996 culminated in Baltag's Structural Theory of Sets STS, which considered

the canonical model of infinitary modal logic as the set theoretical universe.

3 TWO MODELS OF PROVABILITY

According to Brouwer, truth in intuitionistic mathematics means the existence of a proof. An axiom system for intuitionistic logic was suggested by Heyting in 1930; its full description may be found in the fundamental monographs [132, 149, 246]. By IPC, we infer Heyting's intuitionistic propositional calculus. In 1931–34, Heyting and Kolmogorov gave an informal description of the intended proof-based semantics for intuitionistic logic [130, 131, 132, 150], which is now referred to as the *Brouwer-Heyting-Kolmogorov* (*BHK*) *semantics*. According to the *BHK*-conditions, a formula is 'true' if it has a proof. Furthermore, a proof of a compound statement is connected to proofs of its parts in the following way:

- a proof of $A \wedge B$ consists of a proof of proposition A and a proof of proposition B,
- a proof of $A \vee B$ is given by presenting either a proof of A or a proof of B,
- a proof of $A \rightarrow B$ is a construction transforming proofs of A into proofs of B,
- falsehood $\perp$ is a proposition which has no proof; $\neg A$ is shorthand for $A \rightarrow \perp$.

From a foundational point of view, it did not make much sense to understand the above 'proofs' as proofs in an intuitionistic system which those conditions were supposed to

specify. So in 1933 ([109]), Gödel took the first step towards developing an exact semantics for intuitionism based on **classical provability**. Gödel considered the classical modal logic S4 to be a calculus describing properties of provability in classical mathematics:

1. *Axioms and rules of classical propositional logic*,
2. $\Box(F\to G)\to(\Box F\to\Box G)$,
3. $\Box F\to F$,
4. $\Box F\to\Box\Box F$,
5. *Rule of necessitation*: $\dfrac{\vdash F}{\vdash \Box F}$.

Based on Brouwer's understanding of logical truth as provability, Gödel defined a translation $tr(F)$ of the propositional formula F in the intuitionistic language into the language of classical modal logic, i.e., $tr(F)$ was obtained by prefixing every subformula of F with the provability modality $\Box$. Informally speaking, when the usual procedure of determining classical truth of a formula is applied to $tr(F)$, it will test the provability (not the truth) of each of F's subformulas in agreement with Brouwer's ideas.

> Gödel's treatment of provability as modality in [109] has an interesting prehistory. In his letter to Gödel [263] of January 12, 1931, von Neumann actually used formal provability as a modal-like operator B and gave a shorter, modal-style derivation of the second Gödel's incompleteness theorem. Von Neumann freely used such modal logic features as the transitivity axiom $B(a) \to B(B(a))$, equivalent substitution, and the fact that the modality commutes with the conjunction '$\wedge$.' Even earlier, in 1928, Orlov published the paper [205] in Russian, in which he considered an informal modal-like operator of provability, introduced modal postulates (ii)–(v), and described the translation $tr(F)$ from propositional formulas to modal formulas. On the other hand, Orlov chose to base his modal system on a type of relevance logic; his system fell short of S4.

From Gödel's results in [109], and the McKinsey-Tarski work on topological semantics for modal logic [188], it follows that the translation $tr(F)$ provides a proper embedding of the intuitionistic logic IPC into S4, i.e., an embedding of IPC into classical logic extended by the provability operator.

THEOREM 1 (Gödel, McKinsey, Tarski). IPC *proves* F $\Leftrightarrow$ S4 *proves* $tr(F)$.

Still, Gödel's original goal of defining IPC in terms of classical provability was not reached, since the connection of S4 to the usual mathematical notion of provability was not established. Moreover, Gödel noticed that the straightforward idea of interpreting modality $\Box F$ as F *is provable in a given formal system* T contradicted Gödel's second incompleteness theorem (cf. [62, 65, 90, 126, 240] for basic information concerning proof and provability predicates, as well as Gödel's incompleteness theorems).

Indeed, $\Box(\Box F \rightarrow F)$ can be derived in $\mathsf{S4}$ by the rule of necessitation from the axiom $\Box F \rightarrow F$. On the other hand, interpreting modality $\Box$ as the predicate $\mathsf{Provable}_T(\cdot)$ of formal provability in theory T and F as contradiction, i.e., $0 = 1$, converts this formula into a false statement that the consistency of T is internally provable in T:

$$\mathsf{Provable}_T\left(\ulcorner \mathit{Consis(T)} \urcorner\right) .$$

To see this, it suffices to notice that the following formulas are provably equivalent in T:

$$\mathsf{Provable}_T(\ulcorner 0=1 \urcorner) \rightarrow (0=1) \ , \quad \neg\mathsf{Provable}_T(\ulcorner 0=1 \urcorner) \ , \quad \mathit{Consis(T)} \ .$$

Here $\ulcorner\varphi\urcorner$ stands for the Gödel number of φ. Below we will omit Gödel number notation whenever it is safe, e.g., we will write $\mathsf{Provable}(\varphi)$ and $\mathsf{Proof}(n, \varphi)$ instead of $\mathsf{Provable}(\ulcorner\varphi\urcorner)$ and $\mathsf{Proof}(n, \ulcorner\varphi\urcorner)$.

The situation after Gödel's paper [109] can be described by the following figure where '$\hookrightarrow$' denotes a proper embedding:

$$\mathsf{IPC} \ \hookrightarrow \ \mathsf{S4} \ \hookrightarrow \ ? \ \hookrightarrow \ \mathit{CLASSICAL\ PROOFS}\ .$$

In a public lecture in Vienna in 1938 [110], Gödel suggested using the format of explicit proofs *t is a proof of F* for interpreting his provability calculus $\mathsf{S4}$, though he did not give a complete set of principles of the resulting logic of proofs. Unfortunately, Gödel's work [110] remained unpublished until 1995, when the Gödelian logic of proofs had already been axiomatized and supplied with completeness theorems connecting it to both $\mathsf{S4}$ and classical proofs.

The provability semantics of $\mathsf{S4}$ was discussed in [62, 65, 70, 111, 158, 169, 176, 191, 197, 199, 200, 221, 222]. These works constitute a remarkable contribution to this area (cf. Section 4), however, they neither found the Gödelian logic of proofs nor provided $\mathsf{S4}$ with a provability interpretation capable of modeling the *BHK* semantics for intuitionistic logic. Comprehensive surveys of work on provability semantics for $\mathsf{S4}$ may be found in [16, 21, 25].

The Logic of Proofs LP was first reported in 1994 at a seminar in Amsterdam and at a conference in Münster. Complete proofs of the main theorems of the realizability of $\mathsf{S4}$ in LP, and about the completeness of LP with respect to the standard provability semantics were published in the technical report [14] in 1995. The foundational picture now is

$$\mathsf{IPC} \ \hookrightarrow \ \mathsf{S4} \ \hookrightarrow \ \mathsf{LP} \ \hookrightarrow \ \mathit{CLASSICAL\ PROOFS}\ .$$

The correspondence between intuitionistic and modal logics induced by Gödel's translation $tr(F)$ has been studied by Blok, Dummett, Esakia, Flagg, H. Friedman, Grzegorczyk, Kuznetsov, Lemmon, Maksimova, McKinsey, Muravitsky, Rybakov, Shavrukov, Tarski, and many others. A detailed survey of modal companions of intermediate (or superintuitionistic) logics is given in [74]; a brief one is in [75], Sections 9.6 and 9.8.

Gödel's 1933 paper [109] on a modal logic of provability left two natural open problems:

(A) Find a precise provability semantics for the modal logic $\mathsf{S4}$, which appeared to be 'a provability calculus without a provability semantics.'

(B) Find a modal logic of Gödel's predicate of formal provability $\mathsf{Provable}(x)$, which appeared to be 'a provability semantics without a calculus.'

The solution to problem (A) was obtained by Artemov through the Logic of Proofs LP (see above and Section 5).

Problem (B) was solved in 1976 by Solovay, who showed that the modal logic GL (a.k.a. G, L, K4.W, PRL) axiomatized all propositional properties of the provability predicate $\mathsf{Provable}(F)$ ([62, 65, 147, 241, 242]).

The provability logic GL is given by the following list of postulates:

1. *Axioms and rules of classical propositional logic*,
2. $\Box(F\to G)\to(\Box F\to\Box G)$,
3. $\Box(\Box F\to F)\to\Box F$,
4. $\Box F\to\Box\Box F$,
5. *Rule of necessitation*: $\dfrac{\vdash F}{\vdash \Box F}$.

Models (A) and (B) have quite different expressive capabilities. The logic GL formalizes Gödel's second incompleteness theorem $\neg\Box(\neg\Box\bot)$, Löb's theorem $\Box(\Box F\to F)\to\Box F$, and a number of other meaningful provability principles. However, proofs as objects are not present in this model. LP naturally contains typed λ-calculus, modal logic, and modal λ-calculus ([18, 19]). On the other hand, model (A) cannot express Gödel's incompleteness theorem.

Provability models (A) and (B) complement each other by addressing different areas of application. The provability logic GL finds applications in traditional proof theory (cf. Subsection 4.11). The Logic of Proofs LP targets areas of typed theories and programming languages, foundations of verification, formal epistemology, etc. (cf. Subsection 5.7)

4 PROVABILITY LOGIC

A significant step towards finding a modal logic of formal provability was made by Löb who formulated in [180], on the basis of previous work by Hilbert and Bernays from 1939 (see [133]), a number of natural modal-style properties of the formal provability predicate and observed that these properties were sufficient to prove Gödel's second incompleteness theorem. These properties, known as the *Hilbert-Bernays-Löb derivability conditions*, essentially coincide with postulates (2), (4), and (5) of the above formulation of GL, i.e., with the modal logic K4. Moreover, Löb found an important strengthening of the Gödel theorem. He established the validity of the following *Löb Rule* about formal provability:

$$\frac{\vdash \Box F\to F}{\vdash F} \; .$$

It was later noticed in (cf. [182]) that this rule can be formalized in arithmetic, which gave a valid law of formal provability known as *Löb's principle*:

$$\Box(\Box F\to F)\to\Box F \; .$$

This principle provided the last axiom of the provability logic GL, named after Gödel and Löb. Neither Gödel nor Löb formulated the logic explicitly, though they established the validity of the underlying arithmetical principles. Presumably, it was Smiley, whose work [238] on the foundations of ethics was the first to consider GL a modal logic.

Significant progress in the general understanding of the formalization of metamathematics, particularly in [90], inspired Kripke, Boolos, de Jongh, and others to look into the problem of modal axiomatization of the logic of provability. More specifically, the effort was concentrated on establishing GL's completeness with respect to the formal provability interpretation. Independently, a similar problem in an algebraic context was considered by Magari and his school in Italy (see [184]). A comprehensive account of these early developments in provability logic can be found in [66].

H. Friedman formulated the question of decidability of the letterless fragment of provability logic as his Problem 35 in [97]. This question, which happened to be much easier than the general case, was immediately answered by a number of people including Boolos [60], van Benthem, Bernardi, and Montagna. This result was apparently known to von Neumann as early as 1931 [263].

4.1 Solovay's completeness theorem

The modal logic of Gödel's predicate of formal provability $\mathsf{Provable}(x)$ was found in 1976 by Solovay.

Let $*$ be a mapping from the set of propositional letters to the set of arithmetical sentences. We call such a mapping an (arithmetical) *interpretation*. Given a standard provability predicate $\mathsf{Provable}(x)$ in Peano arithmetic PA, we can extend the interpretation $*$ to all modal formulas as follows:

- $\bot^* = \bot$; $\top^* = \top$;
- $*$ commutes with all Boolean connectives;
- $(\Box G)^* = \mathsf{Provable}(G^*)$.

The Hilbert-Bernays-Löb derivability conditions, together with the validity of Löb's principle, essentially mean that GL is sound with respect to the arithmetical interpretation.

PROPOSITION 2. *If* $\mathsf{GL} \vdash X$*, then for all interpretations* $*$, $\mathsf{PA} \vdash X^*$*.*

Solovay in [242] established that GL is also complete with respect to the arithmetical interpretation. Solovay also showed that the set of modal formulas expressing universally *true* principles of provability was axiomatized by a decidable extension of GL, which is usually denoted by S. The system S has the axioms

- all theorems of GL (a decidable set),
- $\Box X \rightarrow X$,

and *modus ponens* as the sole rule of inference.

THEOREM 3 (Solovay [242]).

(1) $\mathsf{GL} \vdash X$ *iff for all interpretations* $*$, $\mathsf{PA} \vdash X^*$*;*

(2) $\mathsf{S} \vdash X$ *iff for all interpretations* $*$, X^* *is true.*

For the proof of this theorem in [242], Solovay invented an elegant technique of embedding Kripke models into arithmetic. Variants and generalizations of this construction have been applied to obtain arithmetical completeness results for various logics with provability and interpretability semantics. An inspection of Solovay's construction shows that it works for all natural formal theories containing a rather weak *elementary arithmetic* EA (cf [25], section 3.1). Such robustness allows us to claim that GL is indeed a universal propositional logic of formal provability.

Whether or not Solovay's theorem can be extended to bounded arithmetic theories such as S^1_2 or S_2 remains an intriguing open question. Interesting partial results here were obtained by Berarducci and Verbrugge in [53].

Solovay's results and methods opened a new page in the development of provability logic. Several groups of researchers in the USA (Solovay, Boolos, Smoryński), the Netherlands (D. de Jongh, Visser), Italy (Magari, Montagna, Sambin, Valentini), and USSR (Artemov and his students), have started to work intensively in this area. An early textbook by Boolos [62], followed by Smoryński's [241], played an important educational role.

The following uniform version of Solovay's Theorem 3.1 was established independently by Artemov, Avron, Boolos, Montagna, and Visser [7, 8, 37, 63, 194, 253]:

there is an arithmetical interpretation $*$ *such that for each modal formula* X, $\mathsf{PA} \vdash X^*$ *iff* $\mathsf{GL} \vdash X$.

The main thrust of the research efforts in the wake of Solovay's Theorem was in the direction of generalizing Solovay's results to more expressive languages. Some of the problems that have received prominent attention are covered below.

4.2 Fixed point theorem

As an important early result on the application of modal logic to the study of the concept of provability in formal systems, a theorem stands out that was found independently by de Jongh and Sambin, who established that GL has the fixed point property (see [62, 65, 240, 241]). The de Jongh-Sambin fixed point theorem is a striking reproduction of Gödel's fixed point lemma in a propositional language free of coding, self-substitution functions, etc.

A modal formula $F(p)$ is said to be *modalized in* p if every occurrence of the sentence letter p in $F(p)$ is within the scope of $\Box$.

THEOREM 4 (de Jongh, Sambin). *For every modal formula* $F(p)$ *modalized in the sentence letter* p*, there is a modal formula* H *containing only sentence letters from* F*, not containing* p*, and such that* GL *proves*

$$H \ \leftrightarrow \ F(H) \ .$$

Moreover, any two solutions to this fixed-point equation with respect to F *are provably equivalent in* GL.

The uniqueness segment was also established by Bernardi in [54].

The proof actually provided an efficient algorithm that, given F, calculates its fixed point H. Here are some examples of F's and their fixed points H.

Modal formula $F(p)$	Its fixed point H
$\Box p$	$\top$
$\Box\neg p$	$\Box\bot$
$\neg\Box p$	$\neg\Box\bot$
$\neg\Box\neg p$	$\bot$
$q \wedge \Box p$	$q \wedge \Box q$

Perhaps the most famous fixed point of the above sort is given by the second Gödel incompleteness theorem. Indeed, consider $\neg\Box p$ as $F(p)$. By the above table, the corresponding fixed point H is $\neg\Box\bot$. Hence GL proves

(1) $\neg\Box\bot \rightarrow \neg\Box(\neg\Box\bot)$.

Since the arithmetical interpretation of $\neg\Box\bot$ for a given theory T is the consistency formula *Consis(T)*, this yields that (1) represents the formalized second Gödel incompleteness theorem:

if T is consistent, then T does not prove its consistency

and that this theorem is provable in T.

The fixed point theorem for GL allowed van Benthem [248] and then Visser [262] to interpret the modal μ-calculus in GL. Together with van Benthem's observation that GL is faithfully interpretable in μ-calculus [248], this relates two originally disjoint research areas.

4.3 First-order provability logics

The natural problem of axiomatizing first-order provability logic was first introduced by Boolos in [62, 64] as the major open question in this area. A straightforward conjecture that the first-order version of GL axiomatizes first-order provability logic was shown to be false by Montagna [196]. A final negative solution was given in papers by Artemov [9] and Vardanyan [252].

THEOREM 5 (Artemov, Vardanyan). *First-order provability logic is not recursively axiomatizable.*

In particular, Artemov showed that the set of the first-order modal formulas that are true under any arithmetical interpretation is not arithmetical. This proof used Tennenbaum's well-known theorem about the uniqueness of the recursive model of Peano arithmetic. Vardanyan showed that the set of first-order modal formulas that are provable in PA under any interpretation is Π^0_2-complete, thus not effectively axiomatizable. Independently but somewhat later, similar results were obtained by McGee in his Ph.D. thesis; they were never published.

Even more dramatically, [11] showed that first-order provability logics are sensitive to a particular formalization of the provability predicate and thus are not robustly defined.

The material on first-order provability logic is extensively covered in a textbook [65] and in a survey [147].

4.4 Intuitionistic provability logic

The question of generalizing Solovay's results from classical theories to intuitionistic ones, such as Heyting arithmetic HA, proved to be remarkably difficult. Visser in [253] found

a number of nontrivial principles of the provability logic of HA. Similar observations were independently made by Gargov and Gavrilenko. In [255], a characterization and a decision algorithm for the letterless fragment of the provability logic of HA were obtained, thus solving an intuitionistic analog of Friedman's 35th problem.

THEOREM 6 (Visser [255]). *The letterless fragment of the provability logic of* HA *is decidable.*

Some significant further results were obtained in [79, 135, 136, 137, 255, 258, 260, 261], but the general problem of axiomatizing the provability logic of HA remains a major open question.

4.5 Classification of provability logics

Solovay's theorems naturally led to the notion of *provability logic for a given theory T relative to a metatheory U*, which was suggested by Artemov in [7, 8] and Visser in [253]. This logic, denoted $\boldsymbol{PL}_T(U)$, is defined as the set of all propositional principles of provability in T that can be established by means of U. In particular, GL is the provability logic $\boldsymbol{PL}_T(U)$ with $U = T = \mathsf{PA}$, and Solovay's provability logic S from Theorem 3.2 corresponds to $T = \mathsf{PA}$ and U's being the set of all true sentences of arithmetic. The problem of describing all provability logics for a given theory T relative to a metatheory U, where T and U range over extensions of Peano arithmetic, has become known as the *classification problem for provability logics*. Each of these logics extends GL, hence can be represented in the form $\mathsf{GL}X$ which is GL with additional axioms X and modus ponens as the sole rule of inference. Within this notational convention, $\mathsf{S}{=}\mathsf{GL}\{\Box p \to p\}$. Consider sentences $F_n = \Box^{n+1}\bot \to \Box^n\bot$, for $n \in \omega$. In [8, 10, 254], the following three families of provability logics were found:

$$\mathsf{GL}_\alpha = \mathsf{GL}\{F_n \mid n \in \alpha\},\ \text{ where } \alpha \subseteq \omega\ ;$$

$$\mathsf{GL}^-_\beta = \mathsf{GL}\{\bigvee_{n \notin \beta} \neg F_n\},\ \text{ where } \beta \text{ is a confinite subset of } \omega\ ;$$

$$\mathsf{S}_\beta = \mathsf{S} \cap \mathsf{GL}^-_\beta,\ \text{ where } \beta \text{ is a confinite subset of } \omega\ .$$

The families GL_α, GL^-_β and S_β are ordered by inclusion of their indices, and $\mathsf{GL}_\beta \subset \mathsf{S}_\beta \subset \mathsf{GL}^-_\beta$, for cofinite β.

In [10], the classification problem was reduced to finding all provability logics in the interval between GL_ω and S. In [143], Japaridze found a new provability logic Dzh in this interval,

$$\mathsf{Dzh} = \mathsf{GL}\{\neg\Box\bot, \Box(\Box p \vee \Box q) \to (\Box p \vee \Box q)\}\ .$$

He showed that Dzh is the provability logic of PA relative to $\mathsf{PA}+$ *formalized ω-consistency of* PA. This discovery produced one more provability logic series,

$$\mathsf{Dzh}_\beta = \mathsf{Dzh} \cap \mathsf{GL}^-_\beta,\ \text{ where } \beta \text{ is a confinite subset of } \omega\ ,$$

with $\mathsf{GL}_\beta \subset \mathsf{Dzh}_\beta \subset \mathsf{S}_\beta \subset \mathsf{GL}^-_\beta$, for cofinite β.

The classification was completed by Beklemishev who showed in [42] that no more provability logics exist.

THEOREM 7 (Beklemishev [42]). *All provability logics occur in* GL_α, GL^-_β, S_β, *and* Dzh_β, *for* $\alpha, \beta \subseteq \omega$, *and* β *cofinite.*

The proof of Theorem 7 produced yet another provability interpretation of Dzh which was shown to be the provability logic of any Σ_1-sound but not sound theory relative to the set of all true sentences of arithmetic. For more details, see [25, 42, 50].

4.6 Provability logics with additional operators

Solovay's theorems have been generalized to various extensions of the propositional language by additional operators having arithmetical interpretations.

The most straightforward generalization is obtained by simultaneously considering several provability operators corresponding to different theories. Already in the simplest case of *bimodal provability logic*, the axiomatization of such logics turns out to be very difficult. The bimodal logics for many natural pairs of theories have been characterized in [43, 44, 73, 143, 241]. However, the general classification problem for bimodal provability logics for pairs of recursively enumerable extensions of PA remains a major open question.

Bimodal logic has been used to study relationships between provability and interesting related concepts such as the Mostowski operator, and Rosser, Feferman, and Parikh provabilities (see [179, 225, 226, 241, 256]). In a number of cases, Solovay-style arithmetical completeness theorems have been obtained. These results have their origin in an important paper by Guaspari and Solovay [123] (see also [241]). They consider an extension of the propositional modal language by a *witness comparison* operator allowing the formalization of Rosser-style arguments from his well-known proof of the incompleteness theorem [218]. Similar logics have since been used in [71, 72, 78], e.g., in the study of the speed-up of proofs.

4.7 Generalized provability predicates

A natural generalization of the provability predicate is given by the notion of n-*provability* which is, by definition, a provability predicate in the set of all true arithmetical Π_n-sentences. For $n = 0$, this concept coincides with the usual notion of provability. As was observed in [241], the logic of each individual n-provability predicate coincides with GL. A joint logic of n-provability predicates for $n = 0, 1, 2, \ldots$ contains the modalities $[0]$, $[1]$, $[2]$, etc. The arithmetical interpretation of a formula in this language is defined as usual, except that now we require, for each $n \in \omega$, that $[n]$ be interpreted as n-provability.

The system GLP introduced by Japaridze [143, 144] is given by the following axioms and rules of inference.

1. *Axioms of* GL *for each operator* $[n]$,
2. $[m]\phi \rightarrow [n]\phi$, *for* $m \leq n$,
3. $\langle m\rangle\phi \rightarrow [n]\langle m\rangle\phi$, *for* $m < n$,
4. *Rule modus ponens*,
5. *Rule* $\phi \vdash [n]\phi$.

THEOREM 8 (Japaridze). GLP *is sound and complete with respect to the n-provability interpretation.*

Originally, Japaridze established in [143, 144] the completeness of GLP for an interpretation of modalities $[n]$ as the provability in arithmetic using not more than n nested applications of the ω-rule. Later, Ignatiev in [141] observed that Japaridze's theorem holds for the n-provability interpretation. Ignatiev also found normal forms for letterless formulas in GLP which play a significant role in Section 4.11 (where only the soundness of GLP is essential).

4.8 Interpretability and conservativity logics

Interpretability is one of the central concepts of mathematics and logic. A theory X is interpretable in Y iff the language of X can be translated into the language of Y in such a way that Y proves the translation of every theorem of X. For example, Peano arithmetic PA is interpretable in Zermelo-Fraenkel set theory ZF. The importance of this concept lies in its ability to compare theories of different mathematical character in different languages, e.g., set theory and arithmetic. The notion of interpretability was given a mathematical shape by Tarski in 1953 in [245]. There is not much known about interpretability in general. The modal logic approach provides insights into the structure of interpretability in special situations when X and Y are finite propositional-style extensions of a base theory containing a certain sufficient amount of arithmetic.

Visser, following Švejdar [243], introduced a binary modality $A \rhd B$ to stand for the arithmetization of the statement

the theory $T + A$ interprets $T + B$.

Interpretations here are understood in the standard sense of Tarski, and are limited to theories T containing a sufficient amount of arithmetic, and to propositional A's and B's. This new modality emulates provability $\Box F$ by $\neg F \rhd \bot$, and thus is more expressive than the ordinary $\Box$. The resulting *interpretability logic* substantially depends on the basis theory T.

The following logic IL is the collection of some basic interpretability principles valid in all reasonable theories: axioms and rules of GL plus

- $\Box(A \to B) \to A \rhd B$,
- $(A \rhd B \wedge B \rhd C) \to A \rhd C$,
- $(A \rhd C \wedge B \rhd C) \to (A \vee B) \rhd C$,
- $A \rhd B \to (\Diamond A \to \Diamond B)$,
- $\Diamond A \rhd A$.

(We assume here that the interpretability modality '$\rhd$' binds stronger than the Boolean connectives.)

For two important classes of theories T, the interpretability logic has been characterized axiomatically.

Let ILP be IL augmented by the principle

$$A \rhd B \rightarrow \Box(A \rhd B) .$$

THEOREM 9 (Visser [257]). *The interpretability logic of a finitely axiomatizable theory satisfying some natural conditions is* ILP.

In particular, the class of theories covered by this theorem includes the arithmetical theories $\mathsf{I}\Sigma_n$ for all $n = 1, 2, 3, \ldots$, the second-order arithmetic ACA_0, and the von Neumann-Gödel-Bernays theory GB of sets and classes.

Let ILM be IL augmented by Montagna's principle

$$A \rhd B \rightarrow (A \wedge \Box C) \rhd (B \wedge \Box C) .$$

The following theorem was established independently in [224] and [52].

THEOREM 10 (Shavrkurov, Berarducci). *The interpretability logic of essentially reflexive theories satisfying some natural conditions is* ILM.

In particular, this theorem states that ILM is the interpretability logic for Peano arithmetic PA and Zermelo-Fraenkel set theory ZF.

An axiomatization of the minimal interpretability logic, i.e., of the set of interpretability principles that hold over all reasonable arithmetical theories, is not known. Important progress in this area has been made by Goris and Joosten, who have found new universal interpretability principles (cf. [120, 148]). Yet more new interpretability principles have been found recently by Goris; they were discovered using the Kripke semantics and later shown sound for arithmetic.

The $\rhd$ modality has a related *conservativity* interpretation, which leads to the conservativity logics studied in [124, 125, 140]. Logics of *interpolability* and of *tolerance*, introduced by Ignatiev and Japaridze [80, 81, 142], have a related arithmetical interpretation, but a format which is different from that of interpretability logics; see [147] for an overview.

An excellent survey of interpretability logic is given in [259]; see also [147].

4.9 Magari algebras and propositional second-order provability logic

An algebraic approach to provability logic was initiated by Magari and his students [183, 184, 194, 195]. The *provability algebra* of a theory T, also called the *Magari algebra of* T, is defined as the set of T-sentences factorized modulo provable equivalence in T and equipped with the usual Boolean operations together with the provability operator mapping a sentence F to $\mathsf{Provable}_T(F)$.

Using the notion of provability algebra, one can impart a provability semantics to a representative subclass of propositional second-order modal formulas, i.e., modal formulas with quantifiers over arithmetical sentences. These are just first-order formulas over the provability algebra. For several years, the questions of decidability of the propositional second-order provability logic and of the first-order theory of the provability algebra of PA remained open (cf. [24]). Shavrukov in [227] provided a negative solution to both of these questions.

THEOREM 11 (Shavrukov [227]). *The first-order theory of the provability algebra of* PA *is mutually interpretable with the set of all true arithmetical formulas.*

This result was proved by one of the most ingenious extensions of Solovay's techniques.

4.10 'True and Provable' modality

A gap between the provability logic GL and S4 can be bridged to some extent by using the *strong provability* modality $\boxdot F$ which is interpreted as

$$(\boxdot F)^* = F^* \wedge \mathsf{Provable}(F^*) \ .$$

The reflexivity principle

$$\boxdot F \rightarrow F$$

is then vacuously provable, hence the strong provability modality is S4-compliant.

This approach has been explored in [61, 111, 170], where it was shown independently that the arithmetically complete modal logic of strong provability coincides with Grzegorczyk's logic Grz, which is the extension of S4 by the axiom

$$\Box(\Box(F \rightarrow \Box F) \rightarrow F) \rightarrow F \ .$$

The modality of strong provability has been further studied in [202, 203]; it played a significant role in introducing justification into formal epistemology (cf. [30, 31, 32]), as well as in the topological semantics for modal logic (cf. surveys [86, 100]).

Strong provability plays a certain foundational role: it provides an exact provability-based model for intuitionistic logic IPC. Indeed, by Grzegorczyk's result from [122], Gödel's translation *tr* specifies an exact embedding of IPC into Grz (cf. Theorem 1):

$$\mathsf{IPC}\ \textit{proves}\ F \quad \Leftrightarrow \quad \mathsf{Grz}\ \textit{proves}\ tr(F)\ .$$

However, the foundational significance of this reduction for intuitionistic logic is somewhat limited by a nonconstructive meaning of strong provability as 'classically true and formally provable,' which seems incompatible with the intended intuitionistic semantics. The aforementioned embedding does not bring us closer to the *BHK* semantics for IPC either. For more discussion on these matters, see [12, 16, 171].

4.11 Applications

The methods of modal provability logic are applicable to the study of fragments of Peano arithmetic.

Using provability logic methods, Beklemishev in [45] answered a well-known question: what kind of computable functions could be proved to be total in the fragment of PA where induction is restricted to Π_2-formulas without parameters? He showed that these functions coincide with those that are primitive recursive. In general, provability logic analysis substantially clarified the behavior of parameter-free induction schemata.

Later results [46, 48] revealed a deeper connection between provability logic and traditional proof-theoretic questions, such as consistency proofs, ordinal analysis, and independent combinatorial principles. In [48], Beklemishev gave an alternative proof of

Gentzen's famous theorem on the proof of the consistency of PA by transfinite induction up to the ordinal ϵ_0.

In [47] (cf. also surveys [25, 49]), Beklemishev suggested a simple PA-independent combinatorial principle called *the Worm Principle*, which is derived from Japaridze's polymodal extension GLP of provability logic (cf. Section 4.7). Finite words in the alphabet of natural numbers will be called *worms.* The Worm Principle asserts the termination of any sequence $w_0, w_1, w_2, \ldots$ of worms inductively constructed according to the following two rules. Suppose $w_m = x_0 \ldots x_n$, then

1. if $x_n = 0$, then $w_{m+1} := x_0 \ldots x_{n-1}$ (the head of the worm is cut away);
2. if $x_n > 0$, set $k := \max\{i < n : x_i < x_n\}$ and let $w_{m+1} = x_0 \ldots x_k(x_{k+1} \ldots x_{n-1}(x_n - 1))^{m+1}$ (the head of the worm decreases by one, and the part after position k is appended to the worm m times).

Clearly, the emerging sequence of worms is fully determined by the initial worm w_0. For example, consider a worm $w_0 = 2031$. Then the sequence looks as follows:

$w_0 = 2031$
$w_1 = 203030$
$w_2 = 20303$
$w_3 = 20302222$
$w_4 = 2030222122212221222122221$
$w_5 = 2030(22212221222122212220)^6$
$\ldots$

THEOREM 12 (Beklemishev [47]).

(1) *For any initial worm w_0, there is an m such that w_m is empty.*

(2) *The previous statement is unprovable in Peano arithmetic* PA. *In fact, Statement 1 is equivalent to the 1-consistency of* PA.

For other PA-independent principles, cf. [244].

Japaridze used a technique from the area of Provability Logic to investigate fundamental connections between provability, computability, and truth in his work on Computability Logic [145, 146].

The Logic of Proofs (Section 5) with its applications also emerged from studies in Provability Logic.

5 LOGIC OF PROOFS

The source of difficulties in provability interpretation of modality lies in the implicit nature of the existential quantifier $\exists$. Consider, for instance, the reflection principle in PA, i.e., all formulas of type $\mathsf{Provable}(F) \to F$. By Gödel's second incompleteness theorem, this principle is not provable in PA, since the consistency formula $\mathsf{Con}(\mathsf{PA})$ coincides with a special case of the reflection principle, namely $\mathsf{Provable}(\bot) \to \bot$. The formula $\mathsf{Provable}(F)$ is $\exists x \mathsf{Proof}(x, F)$ where $\mathsf{Proof}(x, y)$ is Gödel's *proof predicate*

$$x \textit{ is (a code of) a proof of a formula (having code) } y\,.$$

Assuming $\mathsf{Provable}(F)$ does not yield pointing to any specific proof of F, since this x may be a nonstandard natural number which is not a code of any actual derivation in PA.

For proofs represented by explicit terms, the picture is very different. The principle of *explicit reflection* $\mathsf{Proof}(p, F) \rightarrow F$ is provable in PA for each specific derivation p. Indeed, if $\mathsf{Proof}(p, F)$ holds, then F is evidently provable in PA, and so is the formula $\mathsf{Proof}(p, F) \rightarrow F$. Otherwise, if $\mathsf{Proof}(p, F)$ does not hold, then $\neg\mathsf{Proof}(p, F)$ is true and provable, therefore $\mathsf{Proof}(p, F) \rightarrow F$ is also provable.

This observation suggests a remedy: representing proofs by terms t in the proof formula $\mathsf{Proof}(t, F)$ instead of implicit representation of proofs by existential quantifiers in the provability formula $\exists x \mathsf{Proof}(x, F)$. As we have already mentioned, Gödel suggested using the format of explicit proof terms for the interpretation of $\mathsf{S4}$ as early as 1938, but that paper remained unpublished until 1995 ([110]). Independently, the study of explicit modal logics was initiated in [14, 33, 34, 35, 247]. The Logic of Proofs may be regarded an instance of Gabbay's Labelled Deductive Systems (cf. [98]).

Proof polynomials are terms built from *proof variables* $x, y, z, \ldots$ and *proof constants* $a, b, c, \ldots$ by means of three operations: *application* '$\cdot$' (binary), *union* '$+$' (binary), and *proof checker* '!' (unary). The language of *Logic of Proofs* LP is the language of classical propositional logic supplemented by a new rule for building formulas, namely for each proof polynomial p and formula F, there is a new formula $p{:}F$ denoting 'p is a proof of F.' It is also possible to read this language type-theoretically: formulas become types, and $p{:}F$ denotes 'term p has type F.' We assume also that '$t{:}$' and '$\neg$' bind stronger than '$\wedge, \vee$' which, in turn, bind stronger than '$\rightarrow$.'

Axioms and inference rules of LP:

1. *Axioms of classical propositional logic*
2. $t{:}(F \rightarrow G) \ \rightarrow \ (s{:}F \rightarrow (t \cdot s){:}G)$ (*application*)
3. $t{:}F \rightarrow F$ (*reflection*)
4. $t{:}F \ \rightarrow\, !t{:}(t{:}F)$ (*proof checker*)
5. $s{:}F \rightarrow (s+t){:}F, \quad t{:}F \rightarrow (s+t){:}F$ (*sum*)
6. *Rule modus ponens*
7. $\vdash c{:}A$, *where* A *is from* 1-5, *and* c *is a proof constant* (*constant specification*)

As one can see from the principles of LP, constants denote proofs of axioms. The application operation corresponds to the internalized *modus ponens* rule: for each s and t, a proof $s \cdot t$ is a proof of all formulas G such that s is a proof of $F \rightarrow G$ and t is a proof of F for some F. The sum '$s + t$' of proofs s and t is a proof which proves everything that either s or t does. Finally, '!' is interpreted as a universal program for checking the correctness of proofs, which given a proof t, produces a proof that t proves F ([14, 16]). In [17], it was noted that proof polynomials represent the whole set of possible operations on proofs for a propositional language. It was shown that any operation on proofs which is invariant with respect to a choice of a normal proof system and which can be specified in a propositional language can be realized by a proof polynomial.

In what follows, '$\vdash$' denotes derivability in LP unless stated otherwise. By a *constant specification* $\mathcal{CS}$, we mean a set of formulas $\{c_1{:}A_1, c_2{:}A_2, \ldots\}$ where each A_i is an axiom from 1–5 of LP, and each c_i is a proof constant. By default, with each derivation in LP,

we associate a constant specification $\mathcal{CS}$ introduced in this derivation by the use of the rule of constant specification.

One of the basic properties of LP is its capability of internalizing its own derivations. The weak form of this property yields the following admissible rule for LP ([14, 16]):

$$\textit{if} \vdash F,\ \textit{then} \vdash p{:}F\ \textit{for some proof polynomial}\ p\ .$$

This rule is a translation of the well-known necessitation rule of modal logic

$$\frac{\vdash F}{\vdash \Box F}$$

into the language of explicit proofs. The following more general *internalization rule* holds for LP: *if*

$$A_1, \ldots, A_n \vdash B\ ,$$

then it is possible to construct a proof polynomial $t(x_1, \ldots, x_n)$ *such that*

$$x_1{:}A_1, \ldots, x_n{:}A_n \vdash t(x_1, \ldots, x_n){:}B$$

One might notice that the Curry-Howard isomorphism covers only a simple instance of the proof internalization property where all of $A_1, \ldots, A_n, B$ are purely propositional formulas containing no proof terms. For the Curry-Howard isomorphism basics, see, e.g., [108].

The decidability of LP was established by Mkrtychev in [193]. Kuznets in [168] obtained an upper bound Σ_2^p on the satisfiability problem for LP-formulas in Mkrtychev models (cf. Section 5.3). This bound was lower than the known upper bound *PSPACE* on the satisfiability problem in $\mathsf{S4}$ (under the assumption that $\Sigma_2^p \neq \mathit{PSPACE}$). A possible explanation of why LP wins in complexity over $\mathsf{S4}$ is that the satisfiability test for LP is somewhat similar to type checking, i.e., checking the correctness of assigning types (formulas) to terms (proofs), which is known to be relatively easy in classical cases.

Milnikel in [190] established Π_2^p-completeness of LP for some natural classes of constant specifications, including so-called injective ones, when each constant denotes a proof of not more than one axiom. Π_2^p-hardness for the whole LP remains on open problem.

N. Krupski in [159] established the disjunctive property for LP:

$$\textit{if}\ \mathsf{LP} \vdash s{:}F \vee t{:}G,\ \textit{then}\ \mathsf{LP} \vdash s{:}F\ \textit{or}\ \mathsf{LP} \vdash t{:}G.$$

5.1 Arithmetical completeness

The Logic of Proofs LP is sound and complete with respect to the natural provability semantics. By a *proof system* we mean a provably in PA decidable predicate $\mathsf{Proof}(x, y)$ that enumerates all theorems of PA, i.e.,

$$\mathsf{PA} \vdash \varphi \quad \text{iff} \quad \mathsf{Proof}(n, \varphi)\ \text{ holds for some } n\ ,$$

together with computable functions $\mathbf{m}(x, y)$, $\mathbf{a}(x, y)$ and $\mathbf{c}(x)$ which satisfy identities for '$\cdot$,' '$+$,' and '!' respectively, i.e., for all arithmetical formulas φ, ψ and all natural numbers k, n the following holds:

$$\mathsf{Proof}(k, \varphi \to \psi) \wedge \mathsf{Proof}(n, \varphi) \to \mathsf{Prf}(\mathbf{m}(k, n), \psi)$$

$\mathsf{Proof}(k,\varphi)\rightarrow\mathsf{Proof}(\mathbf{a}(k,n),\varphi),\qquad \mathsf{Proof}(n,\varphi)\rightarrow\mathsf{Proof}(\mathbf{a}(k,n),\varphi)$
$\mathsf{Proof}(k,\varphi)\rightarrow\mathsf{Proof}(\mathbf{c}(k),\mathsf{Proof}(k,\varphi))$.

The class of proof systems includes the Gödelian proof predicate in PA

x is a Gödel number of a derivation in PA *containing a formula with a Gödel number y*

with obvious choice of operations $\mathbf{m}(x,y)$, $\mathbf{a}(x,y)$ and $\mathbf{c}(x)$. In particular, $\mathbf{a}(n,m)$ is the concatenation of proofs n and m, and $\mathbf{c}$ is a computable function that given a Gödel number of a proof n, returns the Gödel number $\mathbf{c}(n)$ of a proof, containing formulas $\mathsf{Proof}(n,\varphi)$ for all φ's such that $\mathsf{Proof}(n,\varphi)$ holds.

An arithmetical interpretation $*$ is determined by a choice of proof system as well as an interpretation of proof variables and constants by numerals (denoting proofs), and propositional variables by arithmetical sentences. Boolean connectives are understood in the same way in LP and PA, and a formula $p{:}F$ is interpreted as an arithmetical formula $\mathsf{Proof}(p^*,F^*)$.

> This kind of provability semantics is referred to as *call-by-value semantics*; it was introduced in [15] and used in [16, 18, 29, 119, 270]. A more sophisticated *call-by-name semantics* of the language of LP was introduced in [14] and used in [160, 161, 235, 269]. Under the call-by-name semantics, proof polynomials are interpreted as Gödel numbers of definable provably recursive arithmetical terms. Call-by-value interpretations may be regarded as a special case of call-by-name interpretations since numerals are definable provably recursive arithmetical terms.

For a given constant specification $\mathcal{CS}$, an interpretation $*$ is called a *$\mathcal{CS}$-interpretation* if all formulas from $\mathcal{CS}$ are true under a given $*$. The following arithmetical completeness theorem has been established in [14] for the call-by-name semantics and in [15] for the call-by-value semantics (see also articles [16, 18]):

THEOREM 13 (Artemov [14, 15]). *A formula F is derivable in* LP *with a given constant specification $\mathcal{CS}$ iff* $\mathsf{PA}\vdash F^*$, *for any $\mathcal{CS}$-interpretation $*$.*

This theorem stands if one replaces '$\mathsf{PA}\vdash F^*$' by 'F^* holds in the standard model of arithmetic.'

In his recent paper [119], Goris showed that LP is sound and complete with respect to the call-by-value semantics of proofs in Buss's weak arithmetic S^1_2, thus showing that proof polynomials can be realized by *PTIME*-computable operations on proofs. Note that the corresponding question for the Provability Logic GL remains a major open problem.

The logic of single-conclusion proofs was described by V. Krupski in [160, 161]. This system does not correspond to any normal modal logic.

5.2 Realization Theorem

Another major feature of the Logic of Proofs is its ability to realize all S4-derivable formulas by restoring corresponding proof polynomials inside all occurrences of modality. This fact may be expressed by the following realization theorem ([14, 16]). By a *forgetful projection* of an LP-formula F, we understand a modal formula obtained by replacing all assertions $t{:}(\cdot)$ in F by $\Box(\cdot)$.

THEOREM 14 (Artemov [14]). *$\mathsf{S4}$ is the forgetful projection of LP.*

That the forgetful projection of LP is $\mathsf{S4}$-compliant is a straightforward observation. The converse has been established in [14, 16] by presenting an algorithm which substitutes proof polynomials for all occurrences of modalities in a given cut-free Gentzen-style $\mathsf{S4}$-derivation of a formula F, thereby producing a formula F^r derivable in LP. The original realization algorithms from [14, 16] were exponential. Brezhnev and Kuznets in [69] offered a realization algorithm of $\mathsf{S4}$ into LP which is polynomial in the size of a cut-free derivation in $\mathsf{S4}$. The lengths of realizing proof polynomials can be kept quadratic in the length of the original cut-free $\mathsf{S4}$-derivation.

Here is an example of an $\mathsf{S4}$-derivation realized as an LP-derivation in the style of Theorem 14. There are two columns in the table below. The first is a Hilbert-style $\mathsf{S4}$-derivation of a modal formula $\Box A \vee \Box B \rightarrow \Box(\Box A \vee B)$. The second column displays corresponding steps of an LP-derivation resulted in an LP-proof of a formula

$$x{:}A \vee y{:}B \rightarrow (a{\cdot}!x + b{\cdot}y){:}(x{:}A \vee B)$$

with constant specification

$$\{\ a{:}(x{:}A \rightarrow x{:}A \vee B),\quad b{:}(B \rightarrow x{:}A \vee B)\ \}\ .$$

	Derivation in $\mathsf{S4}$	Derivation in LP
1.	$\Box A \rightarrow \Box A \vee B$	$x{:}A \rightarrow x{:}A \vee B$
2.	$\Box(\Box A \rightarrow \Box A \vee B)$	$a{:}(x{:}A \rightarrow x{:}A \vee B)$
3.	$\Box\Box A \rightarrow \Box(\Box A \vee B)$	$!x{:}x{:}A \rightarrow (a{\cdot}!x){:}(x{:}A \vee B)$
4.	$\Box A \rightarrow \Box\Box A$	$x{:}A \rightarrow !x{:}x{:}A$
5.	$\Box A \rightarrow \Box(\Box A \vee B)$	$x{:}A \rightarrow (a{\cdot}!x){:}(x{:}A \vee B)$
$5'$.		$(a{\cdot}!x){:}(x{:}A \vee B) \rightarrow (a{\cdot}!x + b{\cdot}y){:}(x{:}A \vee B)$
$5''$.		$x{:}A \rightarrow (a{\cdot}!x + b{\cdot}y){:}(x{:}A \vee B)$
6.	$B \rightarrow \Box A \vee B$	$B \rightarrow x{:}A \vee B$
7.	$\Box(B \rightarrow \Box A \vee B)$	$b{:}(B \rightarrow x{:}A \vee B)$
8.	$\Box B \rightarrow \Box(\Box A \vee B)$	$y{:}B \rightarrow (b{\cdot}y){:}(x{:}A \vee B)$
$8'$.		$(b{\cdot}y){:}(x{:}A \vee B) \rightarrow (a{\cdot}!x + b{\cdot}y){:}(x{:}A \vee B)$
$8''$.		$y{:}B \rightarrow (a{\cdot}!x + b{\cdot}y){:}(x{:}A \vee B)$
9.	$\Box A \vee \Box B \rightarrow \Box(\Box A \vee B)$	$x{:}A \vee y{:}B \rightarrow (a{\cdot}!x + b{\cdot}y){:}(x{:}A \vee B)$

Extra steps $5'$, $5''$, $8'$, and $8''$ are needed in the LP case to reconcile different internalized proofs of the same formula: $(a{\cdot}!x){:}(x{:}A \vee B)$ and $(b{\cdot}y){:}(x{:}A \vee B)$. The resulting realization respects Skolem' idea that negative occurrences of existential quantifiers (here over proofs hidden in the modality of provability) are realized by free variables whereas positive occurrences are realized by functions of those variables.

Switching from the provability format to the language of specific witnesses reveals hidden self-referentiality of modal logic, i.e., the necessity of using proof assertions of the form $t{:}F(t)$, where t occurs in the very formula $F(t)$ of which it is a proof. A recent result by Kuznets in [69] shows that self-referentiality is an intrinsic feature of the modal logic approach to provability in general.

THEOREM 15 (Kuznets [69]). *Self-referential constant specifications of the sort $c{:}A(c)$ are necessary for realization of the modal logic $\mathsf{S4}$ in the Logic of Proofs LP.*

In particular, the $\mathsf{S4}$-theorem

$$\neg\Box\neg(S \to \Box S)$$

cannot be realized in LP without self-referential constant specifications of the sort $c{:}A(c)$.

Systems of proof polynomials for other classical modal logics K, $\mathsf{K4}$, D, $\mathsf{D4}$, T were described in [67, 68]. The case of $\mathsf{S5} = \mathsf{S4} + (\neg\Box F \to \Box\neg\Box F)$ was special because of the presence of negative information about proofs and its connections to formal epistemology. The paper by Artemov, Kazakov, and Shapiro [29] introduced a system of proof terms for $\mathsf{S5}$, and established realizability of the logic $\mathsf{S5}$ by these terms, decidability, and completeness of the resulting logic of proofs.

5.3 Fitting Models

The main idea of epistemic semantics for LP can be traced back to Mkrtychev and Fitting. It consists of augmenting Boolean or Kripke models with an *evidence function*, which assigns 'admissible evidence' terms to a statement before deciding its truth value.

Fitting models are defined as follows. A *frame* is a structure (W, R), where W is a non-empty set of *possible worlds* and R is a binary reflexive and transitive *evidence accessibility* relation on W. Given a frame (W, R), a *possible evidence function* $\mathcal{E}$ is a mapping from worlds and proof polynomials to sets of formulas. We can read $F \in \mathcal{E}(u, t)$ as

'F is one of the formulas for which t serves as possible evidence in world u.'

An *evidence function* is a possible evidence function which respects the intended meanings of the operations on proof polynomials, i.e., for all proof polynomials s and t, for all formulas F and G, and for all $u, v \in W$, each of the following hold:

1. *Monotonicity*: uRv implies $\mathcal{E}(u,t) \subseteq \mathcal{E}(v,t)$;
2. *Closure*:
   - *Application*: $F \to G \in \mathcal{E}(u,s)$ and $F \in \mathcal{E}(u,t)$ implies $G \in \mathcal{E}(u, s\cdot t)$;
   - *Inspection*: $F \in \mathcal{E}(u,t)$ implies $t{:}F \in \mathcal{E}(u, !t)$;
   - *Sum*: $\mathcal{E}(u,s) \cup \mathcal{E}(u,t) \subseteq \mathcal{E}(u, s+t)$.

A model is a structure $\mathcal{M} = (W, R, \mathcal{E}, \Vdash)$ where (W, R) is a frame, $\mathcal{E}$ is an evidence function on (W, R), and $\Vdash$ is an arbitrary mapping from sentence variables to subsets of W. Given a model $\mathcal{M} = (W, R, \mathcal{E}, \Vdash)$, the forcing relation $\Vdash$ is extended from sentence variables to all formulas by the following rules. For each $u \in W$:

1. $\Vdash$ respects connectives ($u \Vdash F \wedge G$ iff $u \Vdash F$ and $u \Vdash G$, $u \Vdash \neg F$ iff $u \nVdash F$, etc.);
2. $u \Vdash t{:}F$ iff $F \in \mathcal{E}(u,t)$ and $v \Vdash F$ for every $v \in W$ with uRv.

We consider the modality $\Box$, associated with the evidence accessibility relation R. In this terms, the last item of the above definition can be recast as

$2'$. $u \Vdash t{:}F$ iff $u \Vdash \Box F$ and t is an admissible evidence for F at u.

Mkrtychev models are Fitting models with singleton W's. LP was shown to be sound and complete with respect to both Mkrtychev models ([193]) and Fitting models ([91, 93]). Fitting models were adapted for multi-agent epistemic setting in [20, 30, 32, 92] and became the standard semantics for justification logics.

5.4 Joint logics of proofs and provability

The problem of finding a joint logic of proofs and provability has been a natural next step, since there are principles that can only be formulated in a mixed language of formal provability and explicit proofs. For example, the modal principle of negative introspection $\neg\Box F \rightarrow \Box\neg\Box F$ is not valid in the provability semantics; neither is a purely explicit version of negative introspection $\neg(x:F) \rightarrow t(x):\neg(x:F)$. However, a mixed explicit-implicit principle $\neg(t{:}F) \rightarrow \Box\neg(t{:}F)$ is valid in the standard provability semantics.

The complete joint system of provability and explicit proofs without operations on proof terms, system B, was found in [13]. This system describes those principles that have a pure logical character and do not depend on any specific operations of proofs.

The postulates of B consist of those of GL together with the following new principles:

- A1. $t{:}F \rightarrow F$,
- A2. $t{:}F \rightarrow \Box t{:}F$,
- A3. $\neg t{:}F \rightarrow \Box\neg t{:}F$,
- RR. *Rule of reflection:* $\dfrac{\vdash \Box F}{\vdash F}$.

THEOREM 16 (Artemov [13]). B *is sound and complete with respect to the semantics of proofs and provability in Peano arithmetic.*

The problem of joining two models of provability, GL and LP, into one model can be specified as that of finding an arithmetically complete logic containing postulates of both GL and LP and closed under internalization.

The first solution to this problem was offered by Yavorskaya (Sidon) who found an arithmetically complete system of provability and explicit proofs, LPP, containing both GL and LP (cf. [235, 269]). Along with natural extensions of principles and operations from GL and LP, LPP contains additional operations '$\Uparrow$' and '$\Downarrow$' which were used to secure the internalization property of LPP. The operation '$\Uparrow$' given a proof t of F, returns a proof $\Uparrow t$ of $\mathsf{Provable}(F)$. The operation '$\Downarrow$' takes a proof t of $\mathsf{Provable}(F)$ and returns a proof $\Downarrow t$ of F. The set of postulates of LPP consists of those of GL and LP together with A2, A3, and RR from B, plus two new principles:

- A4. $t{:}F \rightarrow (\Uparrow t){:}\Box F$,
- A5. $t{:}\Box F \rightarrow (\Downarrow t){:}F$.

Finally, Nogina in [30, 201] noticed that operations '$\Uparrow$' and '$\Downarrow$' along with A4 and A5 are in certain sense redundant and offered a simpler system, GLA, which is an arithmetically complete logic in a joint language of GL and LP, containing postulates of both GL and LP, and closed under internalization. The system GLA is presented in [30, 201] by the set of postulates of GL and LP augmented by the principles:

- $t{:}F \to \Box F$,
- $\neg t{:}F \to \Box \neg t{:}F$,
- $t{:}\Box F \to F$.

and *Rule of reflection* RR.

THEOREM 17.

(1) (Yavorskaya (Sidon) [235, 269]). LPP *is sound and complete with respect to the semantics of proofs and provability in Peano arithmetic.*

(2) (Nogina [30, 201]). GLA *is sound and complete with respect to the semantics of proofs and provability in Peano arithmetic.*

It was the system GLA, which served in [30, 32] as a prototype of justification logics (cf. Subsection 5.7).

5.5 Quantified logics of proofs

The arithmetical provability semantics for the logic of proofs may be naturally generalized to first-order language and to the language of LP with quantifiers over proofs. Both possibilities of enhancing the expressive power of LP were investigated and in both cases, axiomatizability questions have been answered negatively.

THEOREM 18.

(1) (Artemov, Yavorskaya (Sidon) [36]). *The first-order logic of proofs is not recursively enumerable.*

(2) (Yavorsky [271]). *The logic of proofs with quantifiers over proofs is not recursively enumerable.*

An interesting decidable fragment of the first-order logic of the standard proof predicate was found in [270].

5.6 Intuitionistic logic of proofs

The problem of building the intuitionistic logic of proofs has two distinct parts. Firstly, one has to answer the question about the propositional logical principles that axiomatize HA-tautologies in the propositional language enriched by atoms *u is a proof of F* without operations on proof terms, i.e. when u is a variable. The resulting basic logic of proofs reflects purely logical principles of the chosen format. Secondly, one has to pick systems of operations on proofs and study the corresponding intuitionistic logics of proofs. The first of the above problems was solved by Artemov and Iemhoff in [27] where the *Basic Intuitionistic Logic of Proofs*, iBLP, was introduced and found to be arithmetically complete with respect to the semantics of proofs in HA. The paper essentially uses technique and results by de Jongh [77], Smoryński [239], de Jongh and Visser's work on a basis for admissible rules in IPC (circa 1991, cf. [137]), Artemov & Strassen [33] and Artemov [13], Ghilardi [106, 107], Iemhoff [136, 138, 139].

The completeness proof presented in [27] is also interesting because it is the first result in this area for constructive theories; the corresponding problem for the provability logic of Heyting arithmetic HA is still open (Section 4.4).

5.7 Applications

Here we will list some conceptual applications of the Logic of Proofs.

1. *Existential semantics for modal logic.* Proof polynomials and LP represent an exact *existential semantics* for mainstream modal logic. Initially, Gödel regarded the modality $\Box F$ as the provability assertion, i.e.,

there exists a proof for F.

Thus, according to Gödel, modality is an informal Σ_1-sentence, i.e., the one which consists of an existential quantifier (here over proofs) followed by a decidable condition. Such an understanding of modality is typical of 'naive' semantics for a wide range of epistemic and provability logics. Nonetheless, before LP was discovered, major modal logics lacked a mathematical semantics of an existential character. The exception to the rule is the arithmetical provability interpretation for the Provability Logic GL, which still cannot be extended to the major modal logics S4 and S5.

Almost 30 years after the first work by Gödel on the subject, a semantics of a *universal* character was discovered for modal logic, namely Kripke semantics. Modality in that semantics is read informally as the sentence:

in each possible situation, F holds.

Such a reading of modality naturally appears in dynamic and temporal logics aimed at describing computational processes, states of which usually form a (possibly branching) Kripke structure. Universal semantics has been playing a prominent role in modal logic. However, it is not the only possible semantical tool in the study and application of modality. The existential semantics of realizability by proof polynomials can also be useful for foundations and application of modal logic. For more discussion on the existential semantics for modal logic, see [22].

2. *Justification Logic.* A major area of application of the Logic of Proofs is epistemology. The books [89, 189] serve as an excellent introduction to the mathematical logic of knowledge.

Plato's celebrated tripartite definition of knowledge as *justified true belief* is generally regarded in mainstream epistemology as a set of necessary conditions for the possession of knowledge. Due to Hintikka, the 'true belief' components have been fairly formalized by means of modal logic and its possible worlds semantics. The remaining 'justification' condition has received much attention in epistemology (cf., for example, [59, 105, 116, 129, 174, 177, 178, 204]), but lacked formal representation. The issue of finding a formal epistemic logic with justification has also been discussed in [247]. Such a logic contains assertions of the form $\Box F$ (*F is known*), along with those of the form $t{:}F$ (*t is a justification for F*). Justification was introduced into formal epistemology in [20, 30, 31, 32] by combining Hintikka-style epistemic modal logic with justification calculi arising from the Logic of Proofs LP. The generic name for this kind of systems is *Justification Logic*.

3. *Logical omniscience problem.* The traditional Hintikka-style modal logic approach to knowledge has the well-known defect of *logical omniscience*, which is an unrealistic feature that an agent knows all logical consequences of his/her assumptions ([87, 88, 134, 198, 206, 207]). Justification Logic addresses the issue of logical omniscience in a natural way. The paper [28] suggests looking at the logical omniscience as a complexity issue

and offers the following Logical Omniscience Test (LOT): an epistemic system E is not logically omniscient if for any valid in E knowledge assertion $\mathcal{A}$ of type '*F is known*' there is a proof of F in E, the complexity of which is bounded by some polynomial in the length of $\mathcal{A}$. The usual epistemic modal logics are logically omniscient (modulo some common complexity assumptions). On the other hand, Justification Logic is logically omniscient w.r.t. the usual (implicit) knowledge and are not logically omniscient w.r.t. the evidence-based knowledge.

4. *Justified Knowledge.* Justification Logic was used in [20, 23] to offer a new approach to *common knowledge*. A modal operator $J\varphi$ for *justified knowledge* introduced in [20, 23] is defined as a forgetful projection of justification assertions $t{:}\varphi$. Hence the intended meaning of $J\varphi$ is

there is an access to an explicit evidence for φ .

In particular, justified knowledge J was shown in [5, 20, 23] to provide a lighter, constructive version of common knowledge and can be used as such in solving specific problems.

6 MODAL LOGIC OF SPACE

The application of modal logic to topology has a rather long history. The idea of a simple 'algebraic calculus' suitable for proving some topological theorems dates back to Kuratowski [163]. A somewhat similar idea was proposed earlier by Riesz in [216]. A. Robinson in [217] put the problem of developing a topological model theory in the same manner as the classical first-order model theory. Classical first-order logic is insufficient for topology because here one usually deals both with points and sets, hence some fragments of second-order logic should be involved. Topological model theory in this style was developed in [95, 96].

The modal logic approach to topology lies within the same mathematical tradition. Modal calculi can also be interpreted in certain weak fragments of second-order logic. However, modal logics of interest are usually decidable and have a good mathematical structure with respect to both model theory and proof theory. All these features bring into topology some specific logical tools and results.

The use of modal logic in topology was initially motivated by Kuratowski's axioms. Let $\mathcal{T} = \langle \boldsymbol{X}, \mathbb{I} \rangle$ be a topological space, where $\boldsymbol{X}$ is a set of points and $\mathbb{I}$ the interior operation. In terms of the interior ($\mathbb{I}$) and Boolean operations, modal topological principles look as follows:

A1. $\mathbb{I}(Y \cap Z) = \mathbb{I}Y \cap \mathbb{I}Z$;
A2. $\mathbb{I}Y = \mathbb{I}\mathbb{I}Y$;
A3. $\mathbb{I}Y \subseteq Y$;
A4. $\mathbb{I}\boldsymbol{X} = \boldsymbol{X}$

Here Y, Z are subsets of $\boldsymbol{X}$. These axioms can be viewed as identities in the language of Boolean algebras with an extra functional symbol $\mathbb{I}$. They define the variety of so-called *topo-Boolean algebras* (a.k.a. *interior algebras* or *closure algebras*, the latter used by McKinsey and Tarski), and in an obvious way every topological space $\mathcal{T}$ corresponds to a topo-Boolean algebra, which is the powerset of $\mathcal{T}$ with the interior (or closure) operation acting on the subsets of $\mathcal{T}$. The above axioms can also be written as propositional modal formulas: Boolean operations should be replaced by the corresponding propositional

connectives, and $\mathbb{I}$ with the modal connective $\Box$. Thus we obtain the following modal axiom schemes:

A1. $\Box(A \wedge B) = \Box A \wedge \Box B$,

A2. $\Box A \to \Box\Box A$,

A3. $\Box A \to A$,

A4. $\Box\top$,

which are the well-known postulates of $\mathsf{S4}$. This property, noticed in the late 1930s by Tarski, and independently by Stone and Tang, is rather surprising because Lewis' original motivation of $\mathsf{S4}$ was purely logical, and Gödel's provability interpretation of $\mathsf{S4}$ was also of a logical character.

The topological interpretation can be modified to fit other modal logics, namely, one can consider *neighborhood frames*. By definition, such a frame $\mathcal{F} = (\boldsymbol{X}, \mathbb{I})$ is a set $\boldsymbol{X}$ together with an operation $\mathbb{I}$ on its subsets. Then U is called a neighborhood of x if $x \in \mathbb{I}U$. Given a valuation φ which sends proposition letters to subsets of X, we can extend it to all modal formulas as follows:

$\varphi(\neg A) = \boldsymbol{X} \setminus \varphi(A)$,

$\varphi(A \wedge B) = \varphi(A) \cap \varphi(B)$,

$\varphi(A \vee B) = \varphi(A) \cup \varphi(B)$,

$\varphi(\Box A) = \mathbb{I}\varphi(A)$.

The same definition can be given in terms of a forcing relation (or the truth at a point): a formula A is called true at w under interpretation φ if $w \in \varphi(A)$; this is also denoted by $w \Vdash A$. Now the above conditions for extending φ can be written as follows:

$w \Vdash \neg A$ iff $w \nVdash A$,

$w \Vdash A \wedge B$ iff ($w \Vdash A$ and $w \Vdash B$),

$w \Vdash A \vee B$ iff ($w \Vdash A$ or $w \Vdash B$),

$w \Vdash \Box A$ iff $\{y \mid y \Vdash A\}$ is a neighborhood of w.

So $\Box A$ can be read as A *is locally true* [220]. A formula A is called valid in $\mathcal{F}$ (notation: $\mathcal{F} \Vdash A$) if $\varphi(A) = \boldsymbol{X}$ for any valuation φ.

The set $\mathbf{L}(\mathcal{F}) := \{A \mid \mathcal{F} \Vdash A\}$ is called the modal logic of $\mathcal{F}$. Logics of this form are called neighborhood (N-) complete. All well-known modal logics are N-complete. Moreover, they can be presented as logics of familiar topological spaces.

The following theorem is a classical result in this area:

THEOREM 19 (McKinsey, Tarski [187]). *Let $\mathcal{M}$ be a separable dense-in-itself metric space. Then $\mathbf{L}(\mathcal{M}) = \mathsf{S4}$.*

In particular, $\mathbf{L}(\mathbb{R}^n) = \mathsf{S4}$, for each $n = 1, 2, 3, \ldots$.

A simplified proof of topological completeness of $\mathsf{S4}$ with respect to the Cantor space was obtained in [192]. Simplified proofs of completeness of $\mathsf{S4}$ with respect to the real line $\mathbb{R}$ were given in [56, 237].

However, there exist N-incomplete modal logics, even among the extensions of $\mathsf{S4}$. Such examples can be found in [103, 104, 228, 233]. This fact is perhaps counter to the naive intuition: it turns out that there exist systems of topo-Boolean identities that do not correspond to any particular topo-Boolean algebra of a topological space — every such algebra satisfies some other identities that are non-derivable from the original system. This is indeed an incompleteness phenomenon at the level of propositional modal logic akin to those in arithmetical theories.

Kripke semantics can be regarded as a particular case of neighborhood semantics. In fact, given a Kripke frame (W, R), one can build the neighborhood frame $\mathbf{N}(W, R) = (W, \mathbb{I})$ where $\mathbb{I}U := \{x \mid R(x) \subseteq U\}$, so that validities in these two frames are the same. Hence, Kripke-completeness implies N-completeness. The converse is not true; there exist topological spaces with Kripke-incomplete modal logics [103, 104, 228, 233].

For topological semantics of first-order modal logic, see Chapter 9 this Handbook. Modal logics of product topologies were studied in [181, 219, 251]. Gabelaia's master's thesis [100] is a very informative source on modal logic and topology.

6.1 Other operators in topological spaces

The modality $\Box$ can be interpreted in topological spaces not only as the interior, but in some other natural ways. There are other known topological operators on sets that are not expressible in terms of Boolean operation and interior, e.g., taking the derived set $d(X)$ which is the set of all limit points of X [164]. It turns out that the corresponding 'derivational modal logics' of natural classes of topological spaces are not among the most popular modal logics, with one noticeable exception: the derivational modal logic of Cantor's scattered topological spaces turned out to be the Provability Logic GL.

DEFINITION 20. Let C be a class of topological spaces. We understand by $\mathbf{Ld}(C)$ the *derivational modal logic of* C, i.e., the set of propositional formulas with the modality $\Diamond$ interpreted as the derived set operator d that hold in all $\mathcal{T}$'s from C. By $\mathsf{wK4}$ ('weak $\mathsf{K4}$') we understand the modal logic $\mathsf{K} + (p \wedge \Box p) \to \Box\Box p$, and by $\mathsf{D4}$ the logic $\mathsf{K4} + \neg\Box\bot$.

By the modality $\Box^+ F$, we mean $F \wedge \Box F$.

THEOREM 21.

(1) (Esakia [83, 85, 86]). $\mathsf{wK4}$ *is the derivational logic of the class of all topological spaces.*

(2) (Shehtman [230]). *For* $n > 1$, $\mathbf{Ld}(\mathbb{R}^n) = \mathsf{D4} + \Box[(p \wedge \Box p) \vee (\neg p \wedge \Box \neg p)] \to \Box p \vee \Box \neg p$.

(3) (Shehtman [232]). $\mathbf{Ld}(\mathbb{R}) = \mathsf{D4} + \Box(\Box^+ F_1 \vee \Box^+ F_2 \vee \Box^+ F_3) \to (\Box\neg F_1 \vee \Box\neg F_2 \vee \Box\neg F_3)$, *where*

$$F_i = p_i \wedge \bigwedge_{j \neq i} \neg p_j \ .$$

Note that the derivational modal logics of $\mathbb{R}$ and $\mathbb{R}^n$ for $n > 1$ are different. Shehtman in [230] also found that derivational modal logics of $\mathbb{Q}$, Cantor's discontinuum $\mathcal{C}$, as well as any 0-dimensional separable dense-in-itself metric space are all equal to $\mathsf{D4}$. Further results on axiomatization and definability of derivational logics can be found in [55].

A topological space is called *scattered* if it has no dense-in-itself non-empty subsets. Let α be an ordinal. We view α as a topological space with its interval topology. Then it is known that every ordinal α is a scattered space ([186]).

THEOREM 22.

(1) (Esakia [84, 86]). GL *is the derivational logic of the class of all scattered spaces.*

(2) (Abashidze [1], Blass [58]). GL *is the derivational logic of* α, *for any specific ordinal* $\alpha \geq \omega^\omega$.

These theorems demonstrate that Gödel's consistency operator *Con(F)*, stating that

F is consistent with Peano arithmetic ,

and Cantor's topological derived set operator d on scattered spaces have the same set of propositional identities.

6.2 Adding the universal modality

Topological spaces may be considered Boolean algebras with several extra operations, and this leads to different polymodal logics. The basic modal language can be expanded by other modal connectives. For example, one can add the universal modality $[\forall]$, with the following interpretation:

$$w \Vdash [\forall]A \quad \text{iff} \quad x \Vdash A \quad \text{holds for any } x \in \boldsymbol{X}\,.$$

The new language is more expressive: in fact, the formula

$$(\mathrm{AC}) \;:=\; [\forall](\Box p \vee \Box\neg p) \to [\forall]p \vee [\forall]\neg p$$

is valid exactly in connected spaces, but connectedness cannot be expressed in the basic language. Moreover, the following analogs of the classical McKinsey–Tarski Theorem 19 hold. Let

$\mathsf{S4U} = \mathsf{S4}(\text{for } \Box) + \mathsf{S5}(\text{for } [\forall]) + [\forall]p \to \Box p$,

$\mathsf{S4UC} = \mathsf{S4U} + (\mathrm{AC})$.

Let also $\mathbf{LU}(C)$ denote the logic of a class C in the expanded language with $\Box$ and $[\forall]$.

THEOREM 23.

(1) (Goranko, Passy [117]). $\mathsf{S4U} = \mathbf{LU}$ *(all topological spaces).*

(2) (Shehtman [231]). *If $\mathcal{X}$ is a separable dense-in-itself metric space, then*

$$\mathbf{LU}(\mathcal{X}) = \mathsf{S4UC}\,.$$

Some refinement of Shehtman's result (2) can be found in [250] and Chapter 9 of this Handbook. It was shown in [117, 231] that S4U and S4UC have the finite model property, and so they are decidable. As for complexity, S4U is known to be *PSPACE*-complete [6].

An interesting feature is that many *mereotopological relations* between spatial regions (such as 'X is disconnected from Y' or 'X is a (non)tangential proper part of Y') arising in geographical information systems and qualitative spatial representation and reasoning can be expressed within S4U. For example, spatial regions of the *region connection calculus* RCC-8 [51, 82, 211, 212] are interpreted as regular closed subsets of a topological space, and hence can be represented by S4U-formulas of the form $\Diamond\Box X$. The binary relations of RCC-8 can be captured using the universal modality, for instance, $[\forall](\Diamond\Box X \to \Diamond\Box Y)$ says that region X is a part of region Y. RCC-8 is *NP*-complete whereas the satisfiability problem for BRCC-8 (RCC-8 with Boolean operations on regions) in the Euclidean spaces is *PSPACE*-complete, that is, of the same complexity as S4U itself ([101, 266]).

6.3 Modal logic of metric spaces

The first paper on modal logic for metric spaces was, perhaps, the McKinsey and Tarski paper [187], though there were no special modalities for distances there. First-order modal logics for metric spaces were considered in [121]. Modal logics containing specific metric modalities

- $\exists^{<a}$ (or $\exists^{\leq a}$) for 'somewhere in the sphere of radius a excluding (or including) the boundary,' where a is a positive rational number;
- $\exists^{<a}_{>b}$ for 'somewhere at distance d with $b < d < a$,' where $b < a$ are positive rational numbers,

were introduced and studied in [166, 267, 268]. In particular, Wolter and Zakharyaschev in [268] introduced the *modal logic of metric and topology*, MT, in the language containing $\Box$ and $[\forall]$, along with the metric modalities $\exists^{<a}$ and $\exists^{\leq a}$.

THEOREM 24 (Wolter, Zakharyaschev [268]).
(1) MT *is decidable and EXPTIME-complete over arbitrary metric spaces.*
(2) MT *is decidable over the one-dimensional Euclidean space* $\mathbb{R}$*.*
(3) MT *over* $\mathbb{R}^2$ *with the Euclidean metric is undecidable.*

For a survey of other results and further research directions cf. [165].

6.4 Modal logic of dynamic topology

One more class of natural mathematical objects, the *topological dynamic systems*, became a subject of modal logic studies. Two independently working groups can be credited for its origin in 1997: one at Stanford (Kremer, Mints, and Rybakov), and one at Cornell (Artemov, Davoren, and Nerode). We will start by observing the results of the latter, since their approach was more general.

The basic model under consideration is a topological dynamic system $\langle \mathcal{T}, f\rangle$ consisting of a topological space $\mathcal{T} = \langle \boldsymbol{X}, \mathbb{I}\rangle$ and a total function f mapping $\boldsymbol{X}$ to $\boldsymbol{X}$. The corresponding bimodal logic consists of good old S4 with its standard topological interpretation in $\langle \boldsymbol{X}, \mathbb{I}\rangle$, together with a unary modality $\bigcirc$ similar to the one called *the next* or *tomorrow* in temporal logic. A temporal logic was first introduced in [208, 209, 210, 264, 265].

The interpretation of the Boolean connectives is set theoretical in $\boldsymbol{X}$, $\Box$ is interpreted as the interior operation $\mathbb{I}$ on $\mathcal{T}$, and $\bigcirc Y$ is interpreted as $f^{-1}Y$, i.e., the inverse image of Y with respect to f. Hence, the interpretation of $\bigcirc$ reflects the idea of the 'next' temporal operator: $\bigcirc Y$ is the set of points of $\boldsymbol{X}$ which will land in Y 'tomorrow,' after f acts on them once.

DEFINITION 25. The basic system S4F of the dynamic topological logic is S4 together with two temporal principles:

$\bigcirc(A \rightarrow B) \rightarrow (\bigcirc A \rightarrow \bigcirc B)$,
$\bigcirc(\neg A) \leftrightarrow \neg\bigcirc A$,
and the Rule of necessitation for $\bigcirc$: $\dfrac{\vdash A}{\vdash \bigcirc A}$.

The expressive power of S4F suffices to capture the Hoare implication $A \rightarrow \bigcirc B$, stating that with a precondition A after action f, the condition B will hold. One of the main motivations for the authors of [26] to introduce and study dynamic topological logic was to devise a logic tool for analysis of classical and hybrid control systems, where the control function is not necessarily continuous. This line of work has been pursued by Davoren in her dissertation [76], and in subsequent works.

Dynamic systems with continuous function f have been given special treatment. The bimodal language of dynamic topological logic naturally expresses continuity via the principle

$$\textbf{Cont}: \qquad \bigcirc\Box A \to \Box\bigcirc A,$$

reflecting the definition of a continuous mapping as one where an inverse image of an open set is open. Consider the logic

$$\mathsf{S4C} = \mathsf{S4F} + \textbf{Cont} \ .$$

THEOREM 26 (Artemov, Davoren, Nerode [26]).
(1) S4F *is sound and complete with respect to the class of all dynamic systems* $\langle \mathcal{T}, f\rangle$,
(2) S4C *is sound and complete with respect to the class of all dynamic systems* $\langle \mathcal{T}, f\rangle$ *where* f *is continuous on* $\mathcal{T}$.

In addition, S4F and S4C enjoy cut-elimination and the finite model property w.r.t. the corresponding class of Kripke models.

It follows from the proof that S4C is also sound and complete w.r.t. continuous dynamic systems with Alexandrov spaces (the topological equivalents of Kripke frames). Slavnov in [236] and independently Kremer and van Benthem (cf. [156, 237]) showed that the analog of the McKinsey-Tarski completeness theorem does not hold here: S4C is not complete with respect to the real topology over $\mathbb{R}$. In [237], the following weaker form of the McKinsey-Tarski theorem for S4C was established by Slavnov:

if F *is not provable in* S4C*, then* F *has a countermodel in* $\mathbb{R}^n$ *for an appropriate* n .

There is no complete axiomatization known for continuous dynamic systems over $\mathbb{R}^n$ for any specific n.

Dynamic systems $\langle \mathcal{T}, f\rangle$ with continuous f became a starting point for [154, 155, 157]. Consider the logic

$$\mathsf{S4}\bigcirc = \mathsf{S4C} + \Box\bigcirc A \to \bigcirc\Box A \ .$$

THEOREM 27 (Kremer, Mints, Rybakov [156, 157]). S4$\bigcirc$ *is sound and complete w.r.t. the following classes of dynamic systems* $\langle \mathcal{T}, f\rangle$*:*
(a) f *is a homeomorphism;*
(b) $\mathcal{T}$ *is an Alexandrov space,* f *is a homeomorphism;*
(c) $\mathcal{T}$ *is a real topology* $\mathbb{R}^n$*,* f *is a homeomorphism;*
(d) $\mathcal{T}$ *is a unit ball* $\mathcal{B}^n$*,* f *is a measure preserving homeomorphism.*

As was shown in [156], S4$\bigcirc$ has the finite model property, hence it is decidable.

The systems S4F, S4C, and S4$\bigcirc$ (along with so-called *temporal-over-topological fragment* of the dynamic topological logic from [156]), basically exhaust the list of known axiomatizability results in dynamic topology. The papers [76, 155, 157] in addition to $\Box$ and $\bigcirc$, consider the S4-type modality *henceforth*, $*$, to be borrowed from temporal logic, with an apparent goal of capturing some asymptotic behavior of the function f in a dynamic system $\langle \mathcal{T}, f\rangle$. The formal topological interpretation φ of $*B$ is

$$\varphi(*B) = \bigcap_{n\geq 0} f^{-n}\varphi(B) \ ,$$

which specifies the set of points $X \subseteq \varphi(B)$ that never leave $\varphi(B)$ under f, f^2, f^3, etc. The dual of $*$ is the modality $\sharp$, such that $\sharp B$ is interpreted as

$$\varphi(\sharp B) = \bigcup_{n \geq 0} f^{-n}\varphi(B) ,$$

which gives the set of points X that are either in $\varphi(B)$, or reach $\varphi(B)$ under at least one of the iterations f, f^2, f^3, etc. This third modality $*$ considerably extends the expressive power of the dynamic topological logic, making it closer to applications in dynamic systems and control theory. However, this expressive power seems to ruin good algorithmic behavior of dynamic topological logic, as is shown in the following theorem.

THEOREM 28 (Konev, Kontchakov, Wolter, Zakharyaschev [151]). *Let $\mathcal{M}$ be one of the following classes of dynamic systems $\langle \mathcal{T}, f\rangle$:*

(a) *f is a homeomorphism;*
(b) *$\mathcal{T}$ is the class of all Alexandrov spaces, f is a homeomorphism;*
(c) *$\mathcal{T}$ is a real topology $\mathbb{R}^n$, f is a homeomorphism;*
(d) *$\mathcal{T}$ is a unit ball $\mathcal{B}^n$, f is a measure preserving homeomorphism.*

Then the set of valid formulas in the language with $\{\Box, \bigcirc, *\}$ that are valid in $\mathcal{M}$ is not recursively enumerable. All these logics are different.

The proof is by reduction of the Post correspondence problem.

In addition, [151] considers logics for dynamic systems $\langle \mathcal{T}, f\rangle$, where $\mathcal{T}$ is a metric space and f an isometric function. The modal operator for topological interior $\Box$ is replaced by distance operators of the form $\exists^{\leq a}$ 'somewhere in the ball of radius a,' for a positive rational a. In contrast to the topological case, the resulting logic turns out to be decidable, but not bounded in time by any elementary function.

A follow-up paper [102] showed (using more general results on products of modal logics with expanding domains) that the dynamic topological logic interpreted in topological spaces with *continuous* functions was decidable if the number of function iterations was assumed to be *finite*, however, not in primitive recursive time. The decidability proof was based on Kruskal's tree theorem, and the proof of non-primitive recursiveness was established by reduction of the reachability problem for lossy channel systems. Note that the dynamic topological logics interpreted in topological spaces with finite iterations of homeomorphisms are not recursively enumerable.

Quite recently, by encoding the ω-reachability problem for lossy channel systems it was shown in [152] that the dynamic topological logic over some natural spaces with continuous functions is undecidable.

THEOREM 29 (Konev, Kontchakov, Wolter, Zakharyaschev [152]). *The set of formulas in the language with $\{\Box, \bigcirc, *\}$ that are valid in any of the following classes:*

(a) *all continuous dynamic systems $\langle \mathcal{T}, f\rangle$,*
(b) *continuous dynamic systems $\langle \mathcal{T}, f\rangle$ where $\mathcal{T}$ is the class of all Alexandrov spaces,*
(c) *continuous dynamic systems $\langle \mathcal{T}, f\rangle$ where $\mathcal{T}$ is a real topology $\mathbb{R}^n$, is undecidable. All these logics are different.*

This gives a solution to one of the major open problems in the area.

The remaining challenging open questions here are:

1. the decidability and axiomatizability of the dynamic topological logic in the language with $\{\Box, \bigcirc\}$ for the class of continuous dynamic systems over real topological spaces $\mathbb{R}_n$ for fixed $n = 1, 2, 3, \ldots$;

2. the axiomatizability of the dynamic topological logic in the language with $\{\Box, \bigcirc, *\}$ for the class of all continuous dynamic systems.

6.5 Other geometric notions

A number of other fundamental geometrical notions have been connected to corresponding extensions of modal logic in [2, 3, 4]. The paper [3] considered different topological and geometric structures such as connectedness, affine structure, convexity, etc., and proposed a number of languages extending the usual modal language in order to describe these structures. Some authors studied modal logics of such geometric notions as incidence, parallelism, orthogonality, and such structures as projective and affine planes. Precise references and details can be found, e.g., in [38].

The *logic of comparative similarity*, CSL, with the sole metric operator $\leftleftarrows$ for '*closer*' was introduced and investigated in [234]: $X \leftleftarrows Y$ is the set of all points of a given metric space that are closer to set X than to set Y. Despite its apparent simplicity, this langauge is quite impressive. In particular, the topological interior and closure operators as well as the universal modality can be expressed in terms of $\leftleftarrows$.

In all, the above papers contributed to making spatial and spatio-temporal reasoning a lively and actively developing area. Once again, we will refer the reader to the forthcoming collection 'The Logic of Space,' edited by Aiello, van Benthem, and Pratt-Hartmann.

6.6 Modal logic of spacetime

The Minkowski spacetime, together with the causal ($\prec$) and chronological ($\preceq$) accessibility relations, constitute Kripke-style frames which naturally have corresponding modal logics. Knowing such modal logics provide additional understanding of Minkowski's spacetime that forms the basis of Einstein's special theory of relativity. The mathematical problem of finding modal logics for chronological future modality was solved by Goldblatt [112] and Shehtman [229]; the modal logic of the chronological relation $\preceq$ turned out to be $\mathsf{S4.2}{=}\mathsf{S4}{+}\Diamond\Box F \to \Box\Diamond F$. A similar problem for causal future modality was solved by Shapirovsky and Shehtman in [223].

6.7 Topoi

Yet another incarnation of the topological semantics is given by interpreting intuitionistic modality in Grothendieck topology on a category and sheaf theory. Such an interpretation was suggested by Lawvere [173]; a relevant axiomatic system was suggested by Goldblatt in [113]. See the survey [115] and Chapter 9 of this Handbook for exact formulations and discussion. For a different connection between modalities and topos theory relying on geometric morphisms, also see Chapter 9 of this Handbook. An interesting topos-theoretic approach to modality can be found in the works of Reyes and his collaborators [172, 185, 213, 214, 215].

6.8 Universal algebra

A new research thrust in which using modal logic on classical mathematical structures makes a good sense was suggested by Goranko and Vakarelov in [118]. They have devel-

oped a uniform approach to axiomatizing various classes of traditional algebraic structures in modal logic, using the fact that difference modality is naturally definable there.

7 MODALITIES IN SET THEORY

We start with two theorems by Solovay, both published in [65], Chapter 13. These theorems gave a modal characterization of the notions of truth in all transitive models of ZF and truth in all models V_κ, where κ is inaccessible.

Let φ be a function that assigns to each propositional letter a sentence of the language of set theory. For each modal formula A, we define its interpretation, $\varphi(A)$ as follows: φ commutes with Boolean connectives, and $\varphi(\Box A)$ is the sentence of ZF that translates '$\varphi(A)$ holds in all transitive models of ZF.'

Let I be the system of modal logic that results when the principle

$$\Box(\Box A \to \Box B) \vee \Box(\Box B \to (A \wedge \Box A))$$

is added to GL as a new axiom schema.

A *universe* is a set V_κ, where κ is inaccessible. All such V_κ's are models of ZF (cf. [162]). Let ψ be defined as φ before, except that we now define $\psi(\Box A)$ as the sentence of ZF that translates '$\psi(A)$ holds in all universes.' Let J be GL plus the principle

$$\Box(\Box A \to B) \vee \Box((B \wedge \Box B) \to A)\ .$$

THEOREM 30 (Solovay, cf. [65]).

(1) $\mathsf{I} \vdash A$ *iff* $\mathsf{ZF} \vdash \varphi(A)$*, for all* φ *that translate* $\Box A$ *as 'A holds in all transitive models of* ZF*.'*

(2) $\mathsf{J} \vdash A$ *iff* $\mathsf{ZF} \vdash \psi(A)$*, for all* ψ *that translate* $\Box A$ *as 'A holds in all universes.'*

A strong connection between modal logic and non-well-founded sets has been provided by Barwise and Moss in [41] and Baltag in [39, 40]. Suppose one takes ordinary modal logic over some fixed set of atomic sentences and then considers the full infinitary propositional language generated by this. The resulting language has conjunctions and disjunctions of all sets of sentences, and this itself is a proper class of sentences. In addition to this, one can also consider the language with Boolean combinations of at most κ sentences, where κ is a cardinal number. A *pointed model* is a Kripke model with a distinguished point. Bisimulations between pointed models are ordinary bisimulations which relate the distinguished points. Barwise and Moss proved that for every pointed graph (X, x), there is a single sentence $\phi_{X,x}$ which characterizes (X, x) in the sense that for all (Y, y), $(Y, y) \models \phi_{X,x}$ iff (Y, y) is bisimilar to (X, x). The countable case of the Barwise-Moss result had been proved earlier in [249]. It also has roots in infinitary model theory: the Scott sentences there are essentially the same as the characterizing sentences for modal logic. The reason why these results are of interest in non-well-founded set theory is that one way to think about non-well-founded sets is as equivalence classes of pointed models, where the equivalence relation is just the maximum bisimulation. Incidentally, the presence of atomic sentences in the various modal logics then corresponds to the presence of *urelements* in the various set theories.

Viewing the canonical model as a structure for set theory will not give anything like a model of standard ZF because it would have a universal set. However, one can use the

model to obtain a new set theory. This is what Baltag did in [39, 40]. His system STS (Structural Theory of Sets) contains a strengthening of Aczel's AFA axiom, expressed in terms of modal descriptions. Baltag's axiom SAFA, Super-Antifoundation, implies that every maximally consistent class in the infinitary modal logic characterizes some set. STS also has applications to paradoxes and to the 'large/small' distinction in set theory.

Fitting and Smullyan's book [94] is a development of forcing used in independence results, presented in the language of modal logic. The authors use modal terms to explicate many of the combinatorial issues in forcing. Forcing is not usually presented in this way, although it seems quite natural to do so, and they explore a number of affinities between modal logic and forcing.

The paper [58] by Blass presents a set theoretical interpretation of possibility and necessity, based on infinite combinatorics. This is set theoretically meaty, and the focus is on consistency results for infinite combinatorics.

Hamkins' paper [127] introduces the forcing interpretation of modal logic. The focus of the paper, however, is on the Maximality Principle, and it does not use much modal logic beyond observing that the Maximality Principle is equivalent to S5 under the forcing interpretation. The Ph.D. dissertation of Hamkins' student Leibman [175] explores the forcing interpretation of modal logic a bit further. In a recent paper [128] by Hamkins and Löwe, it was proved that the ZFC-provable modal validities for this interpretation are exactly S4.2. There are a large number of open questions in this area.

ACKNOWLEDGEMENTS

We would like to thank Lev Beklemishev, Guram Bezhanishvili, Silvio Ghilardi, Valentin Goranko, Evan Goris, Joel Hamkins, Giorgi Japaridze, Roman Kontchakov, Larry Moss, Elena Nogina, Valentin Shehtman, Johan van Benthem, Rineke Verbrugge, and Michael Zakharyaschev for many valuable suggestions and contributions. Special thanks to Karen Kletter for proofreading and editing this paper.

BIBLIOGRAPHY

[1] Abashidze, M., "Algebraic analysis of the Gödel-Löb modal system," Ph.D. thesis, Tbilisi State University (1987), in Russian.

[2] Aiello, M., "Spatial reasoning: theory and practice," Ph.D. thesis, University of Amsterdam (2002), ILLC Dissertation series, no. 2002-02.

[3] Aiello, M. and J. van Benthem, *A modal walk through space*, Journal of Applied Non-Classical Logics **12** (2002), pp. 319–364.

[4] Aiello, M., J. van Benthem and G. Bezhanishvili, *Reasoning about space: the modal way*, Journal of Logic and Computation **13** (2003), pp. 889–920.

[5] Antonakos, E., *Justified knowledge is sufficient*, Technical Report TR-2006004, CUNY Ph.D. Program in Computer Science (2006).

[6] Areces, C., P. Blackburn and M. Marx, *The computational complexity of hybrid temporal logics*, Logic Journal of IGPL **8** (2000), pp. 653–679.

[7] Artemov, S., "Extensions of arithmetic and modal logics," Ph.D. thesis, Moscow University - Steklov Mathematical Insitute, Moscow (1979), in Russian.

[8] Artemov, S., *Arithmetically complete modal theories*, in: *Semiotika i Informatika*, 14, VINITI, Moscow, 1980 pp. 115–133, in Russian. English translation in: S. Artemov, et al., "Six Papers in Logic," Amer. Math. Soc. Translations (2), volume 135, 1987.

[9] Artemov, S., *Nonarithmeticity of truth predicate logics of provability*, *Soviet Mathematics Doklady* **32** (1985), pp. 403–405.

[10] Artemov, S., *On modal logics axiomatizing provability*, Izvestiya Akademii Nauk SSSR, ser. mat. **49** (1985), pp. 1123–1154, in Russian. English translation in: *Math. USSR Izvestiya* **27** (1986), pp. 401–429.

[11] Artemov, S., *Numerically correct provability logics*, *Soviet Mathematics Doklady* **34** (1987), pp. 384–387.

[12] Artemov, S., *Kolmogorov logic of problems and a provability interpretation of intuitionistic logic*, in: *Theoretical Aspects of Reasoning about Knowledge - III Proceedings* (1990), pp. 257–272.

[13] Artemov, S., *Logic of proofs*, Annals of Pure and Applied Logic **67** (1994), pp. 29–59.

[14] Artemov, S., *Operational modal logic*, Technical Report MSI 95-29, Cornell University (1995).

[15] Artemov, S., *Logic of Proofs: a unified semantics for modality and λ-terms*, Technical Report CFIS 98-06, Cornell University (1998).

[16] Artemov, S., *Explicit provability and constructive semantics*, The Bulletin of Symbolic Logic **7** (2001), pp. 1–36.

[17] Artemov, S., *Operations on proofs that can be specified by means of modal logic*, in: M. Zakharyaschev, K. Segerberg, M. de Rijke and H. Wansing, editors, *Advances in Modal Logic. Volume 2*, CSLI Lecture Notes **119**, CSLI Publications, Stanford, 2001 pp. 59–72.

[18] Artemov, S., *Unified semantics for modality and λ-terms via proof polynomials*, in: K. Vermeulen and A. Copestake, editors, *Algebras, Diagrams and Decisions in Language, Logic and Computation*, CSLI Publications, Stanford University, 2002 pp. 89–119.

[19] Artemov, S., *Embedding of modal λ-calculus into the Logic of Proofs*, Proceedings of the Steklov Mathematical Institute **242** (2003), pp. 36–49.

[20] Artemov, S., *Evidence-based common knowledge*, Technical Report TR-2004018, CUNY Ph.D. Program in Computer Science (2004).

[21] Artemov, S., *Kolmogorov and Gödel's approach to intuitionistic logic: current developments*, Russian Mathematical Surveys **59** (2004), pp. 203–229.

[22] Artemov, S., *Existential semantics for modal logic*, in: S. Artemov, H. Barringer, A. d'Avila Garcez, L. Lamb and J. Woods, editors, *We Will Show Them: Essays in Honour of Dov Gabbay. Volume 1*, College Publications, London, 2005 pp. 19–30.

[23] Artemov, S., *Justified Common Knowledge*, Theoretical Computer Science **357** (2006), pp. 4–22.

[24] Artemov, S. and L. Beklemishev, *On propositional quantifiers in provability logic*, Notre Dame Journal of Formal Logic **34** (1993), pp. 401–419.

[25] Artemov, S. and L. Beklemishev, *Provability Logic*, in: D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, 2nd ed.*, Springer, Dordrecht, 2004 pp. 189–360.

[26] Artemov, S., J. Davoren and A. Nerode, *Modal logics and topological semantics for hybrid systems*, Technical Report MSI 97-05, Cornell University (1997).

[27] Artemov, S. and R. Iemhoff, *The basic intuitionistic logic of proofs*, Technical Report TR-2005002, CUNY Ph.D. Program in Computer Science (2005), to appear in the Journal of Symbolic Logic.

[28] Artemov, S. and R. Kuznets, *Logical omniscience via proof complexity*, Technical Report TR-2006005, CUNY Ph.D. Program in Computer Science (2006), accepted to CSL 2006.

[29] Artemov, S., E. Kazakov and D. Shapiro, *Epistemic logic with justifications*, Technical Report CFIS 99-12, Cornell University (1999).

[30] Artemov, S. and E. Nogina, *Logic of knowledge with justifications from the provability perspective*, Technical Report TR-2004011, CUNY Ph.D. Program in Computer Science (2004).

[31] Artemov, S. and E. Nogina, *Introducing justification to epistemic logic*, Journal of Logic and Computation **15** (2005), pp. 1059–1073.

[32] Artemov, S. and E. Nogina, *On epistemic logic with justification*, in: R. van der Meyden, editor, *Theoretical Aspects of Rationality and Knowledge. Proceedings of the Tenth Conference (TARK 2005), June 10-12, 2005, Singapore* (2005), pp. 279–294.

[33] Artemov, S. and T. Strassen, *The basic logic of proofs*, in: E. Börger, G. Jäger, H. K. Büning, S. Martini and M. Richter, editors, *Computer Science Logic. 6th Workshop, CSL'92. San Miniato, Italy, September/October 1992. Selected Papers*, Lecture Notes in Computer Science **702** (1992), pp. 14–28.

[34] Artemov, S. and T. Strassen, *Functionality in the basic logic of proofs*, Technical Report IAM 93-004, Department of Computer Science, University of Bern, Switzerland (1993).

[35] Artemov, S. and T. Strassen, *The logic of the Gödel proof predicate*, in: G. Gottlob, A. Leitsch and D. Mundici, editors, *Computational Logic and Proof Theory. Third Kurt Gödel Colloquium, KGC'93. Brno, Czech Republic, August 1993. Proceedings*, Lecture Notes in Computer Science **713** (1993), pp. 71–82.

[36] Artemov, S. and T. Yavorskaya (Sidon), *On the first-order logic of proofs*, Moscow Mathematical Journal **1** (2001), pp. 475–490.

[37] Avron, A.,, *On modal systems having arithmetical interpretations*, The Journal of Symbolic Logic **49** (1984), pp. 935–942.

[38] Balbiani, P. and V. Goranko, *Modal logics for parallelism, orthogonality, and affine geometries*, Journal of Applied Non-classical Logics, Special issue on Spatial Reasoning **12** (2002), pp. 365–397.

[39] Baltag, A., "STS: A Structural Theory of Sets," Ph.D. thesis, Indiana University, Bloomington (1998).

[40] Baltag, A., *STS: a structural theory of sets*, in: M. Zakharyaschev, K. Segerberg, M. de Rijke and H. Wansing, editors, *Advances in Modal Logic. Volume 2*, CSLI Lecture Notes **119**, CSLI Publications, Stanford, 2001 pp. 1–34.

[41] Barwise, J. and L. Moss, "Vicious Circles," CSLI publications, Stanford, 1996.

[42] Beklemishev, L., *On the classification of propositional provability logics*, Izvestiya Akademii Nauk SSSR, ser. mat. **53** (1989), pp. 915–943, in Russian. English translation in *Math. USSR Izvestiya* **35** (1990), pp. 247–275.

[43] Beklemishev, L., *On bimodal logics of provability*, Annals of Pure and Applied Logic **68** (1994), pp. 115–160.

[44] Beklemishev, L., *Bimodal logics for extensions of arithmetical theories*, The Journal of Symbolic Logic **61** (1996), pp. 91–124.

[45] Beklemishev, L., *Parameter free induction and provably total computable functions*, Theoretical Computer Science **224** (1999), pp. 13–33.

[46] Beklemishev, L., *Proof-theoretic analysis by iterated reflection*, Arch. Math. Logic **42** (2003), pp. 515–552.

[47] Beklemishev, L., *The Worm principle*, Logic Group Preprint Series 219, University of Utrecht (2003).

[48] Beklemishev, L., *Provability algebras and proof-theoretic ordinals, I*, Annals of Pure and Applied Logic **128** (2004), pp. 103–123.

[49] Beklemishev, L., *Reflection principles and provability algebras in formal arithmetic*, Russian Mathematical Surveys, **60** (2005), pp. 197–268.

[50] Beklemishev, L., M. Pentus and N. Vereshchagin, *Provability, complexity, grammars*, American Mathematical Society Translations, Series 2 **192** (1999).

[51] Bennet, B., *Modal logics for qualitative spatial reasoning*, Logic Journal of the IGPL **4** (1996), pp. 23–45.

[52] Berarducci, A., *The interpretability logic of Peano arithmetic*, The Journal of Symbolic Logic **55** (1990), pp. 1059–1089.

[53] Berarducci, A. and R. Verbrugge, *On the provability logic of bounded arithmetic*, Annals of Pure and Applied Logic **61** (1993), pp. 75–93.

[54] Bernardi, C., *The uniqueness of the fixed point in every diagonalizable algebra*, Studia Logica **35** (1976), pp. 335–343.

[55] Bezhanishvili, G., L. Esakia and D. Gabelaia, *Some results on modal axiomatization and definability for topological spaces*, Studia Logica **81** (2005), pp. 325–355.

[56] Bezhanishvili, G. and M. Gehrke, *Completeness of* $\mathsf{S4}$ *with respect to the real line: revizited*, Annals of Pure and Applied Logic **131** (2005), pp. 287–301.

[57] Blackburn, P., M. de Rijke and Y. Venema, "Modal Logic," Cambridge University Press, 2001.

[58] Blass, A., *Infinitary combinatorics and modal logic*, The Journal of Symbolic Logic **55** (1990), pp. 761–778.

[59] Bonjour, L., *The coherence theory of empirical knowledge*, Philosophical Studies **30** (1976), pp. 281–312, Reprinted in *Contemporary Readings in Epistemology*, M.F. Goodman and R.A. Snyder (eds). Prentice Hall, pp. 70–89, 1993.

[60] Boolos, G., *On deciding the truth of certain statements involving the notion of consistency*, The Journal of Symbolic Logic **41** (1976), pp. 779–781.

[61] Boolos, G., *Reflection principles and iterated consistency assertions*, The Journal of Symbolic Logic **44** (1979), pp. 33–35.

[62] Boolos, G., "The Unprovability of Consistency: An Essay in Modal Logic," Cambridge University Press, 1979.

[63] Boolos, G., *Extremely undecidable sentences*, The Journal of Symbolic Logic **47** (1982), pp. 191–196.

[64] Boolos, G., *The logic of provability*, American Mathematical Monthly **91** (1984), pp. 470–480.

[65] Boolos, G., "The Logic of Provability," Cambridge University Press, 1993.

[66] Boolos, G. and G. Sambin, *Provability: the emergence of a mathematical modality*, Studia Logica **50** (1991), pp. 1–23.

[67] Brezhnev, V., *On explicit counterparts of modal logics*, Technical Report CFIS 2000-05, Cornell University (2000).

[68] Brezhnev, V., *On the Logic of Proofs*, in: *Proceedings of the Sixth ESSLLI Student Session*, Helsinki, 2001 pp. 35–46.

[69] Brezhnev, V. and R. Kuznets, *Making knowledge explicit: How hard it is*, Theoretical Computer Science **357** (2006), pp. 23–34.

[70] Buss, S., *The modal logic of pure provability*, Notre Dame Journal of Formal Logic **31** (1990), pp. 225–231.

[71] Carbone, A. and F. Montagna, *Rosser orderings in bimodal logics*, Zeitschrift f. math. Logik und Grundlagen d. Math. **35** (1989), pp. 343–358.

[72] Carbone, A. and F. Montagna, *Much shorter proofs: a bimodal investigation*, Zeitschrift f. math. Logik und Grundlagen d. Math. **36** (1990), pp. 47–66.

[73] Carlson, T., *Modal logics with several operators and provability interpretations*, Israel Journal of Mathematics **54** (1986), pp. 14–24.

[74] Chagrov, A. and M. Zakharyaschev, *Modal companions of intermediate propositional logics*, Studia Logica **51** (1992), pp. 49–82.

[75] Chagrov, A. and M. Zakharyaschev, "Modal Logic," Oxford Science Publications, 1997.

[76] Davoren, J., "Modal Logics for Continuous Dynamics," Ph.D. thesis, Cornell University (1998).

[77] de Jongh, D., *The maximality of the intuitionistic predicate calculus with respect to Heyting's arithmetic*, The Journal of Symbolic Logic **36** (1970), pp. 606.

[78] de Jongh, D. and F. Montagna, *Much shorter proofs*, Z. Math. Logik Grundlagen Math. **35** (1989), pp. 247–260.

[79] de Jongh, D. and A. Visser, *Embeddings of Heyting algebras*, in: W. Hodges, M. Hyland, C. Steinhorn and J. Truss, editors, *Logic: From Foundations to Applications. European Logic Colloquium, Keele, UK, July 20–29, 1993*, Clarendon Press, Oxford, 1996 pp. 187–213.

[80] Dzhaparidze (Japaridze), G., *The logic of linear tolerance*, Studia Logica **51** (1992), pp. 249–277.

[81] Dzhaparidze (Japaridze), G., *A generalized notion of weak interpretability and the corresponding modal logic*, Annals of Pure and Applied Logic **61** (1993), pp. 113–160.

[82] Egenhofer, M. and R. Franzosa, *Point-set topological spatial relations*, International Journal of Geographical Information Systems **5** (1991), pp. 161–174.

[83] Esakia, L., *The modal logic of topological spaces*, Technical report, The Georgian Academy of Sciences (1976), (in Russian).

[84] Esakia, L., *Diagonal constructions, Löb's formula, and Cantor's scattered spaces*, in: *Logical and semantical investigations*, Tbilisi, Mecniereba, 1981 pp. 128–143, (in Russian).

[85] Esakia, L., *Weak transitivity – a restitution*, in: *Logical Studies, volume 8*, Academic Press, Moscow, 2001 pp. 244–255, (in Russian).

[86] Esakia, L., *Intuitionistic logic and modality via topology*, Annals of Pure and Applied Logic **127** (2004), pp. 155–170, in: *Provinces of logic determined. Essays in the memory of Alfred Tarski. Parts IV, V and VI*, Z. Adamowicz, S. Artemov, D. Niwinski, E. Orlowska, A. Romanowska, and J. Wolenski (eds).

[87] Fagin, R. and J. Halpern, *Belief, awareness and limited reasoning*, in *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, 1985 pp. 480–490.

[88] Fagin, R. and J. Halpern, *Belief, awareness and limited reasoning*, Artificial Intelligence **34** (1988), pp. 39–76.

[89] Fagin, R., J. Halpern, Y. Moses and M. Vardi, "Reasoning About Knowledge," MIT Press, 1995.

[90] Feferman, S., *Arithmetization of metamathematics in a general setting*, Fundamenta Mathematicae **49** (1960), pp. 35–92.

[91] Fitting, M., *A semantics for the Logic of Proofs*, Technical Report TR-2003012, CUNY Ph.D. Program in Computer Science (2003).

[92] Fitting, M., *Semantics and tableaus for* LPS4, Technical Report TR-2004016, CUNY Ph.D. Program in Computer Science (2004).

[93] Fitting, M., *The Logic of Proofs, semantically*, Annals of Pure and Applied Logic **132** (2005), pp. 1–25.

[94] Fitting, M. and R. Smullyan, "Set Theory and the Continuum Problem," Oxford University Press, 1996.

[95] Flum, J., *Model theory of topological structures*, in: M. Krynicki and M. Mostowski and L.W. Szczerba, editor, *Quantifiers: Logics, Models and Computation, vol. 1*, Kluwer Academic Publishers, 1995 pp. 297–312.

[96] Flum, J. and M. Ziegler, "Topological model theory," Lecture Notes in Mathematics **769**, Springer-Verlag, Berlin, 1980.

[97] Friedman, H., *102 problems in mathematical logic*, The Journal of Symbolic Logic **40** (1975), pp. 113–129.

[98] Gabbay, D., "Labelled Deductive Systems," Oxford University Press, 1994.

[99] Gabbay, D., A. Kurucz, F. Wolter and M. Zakharyaschev, "Multi-Dimentional Modal Logics: Theory and Applications," Studies in Logic and the Foundations of Mathematics **148**, Elsevier, Amsterdam, 2003.

[100] Gabelaia, D., *Modal definability in topology* (2001), ILLC Publications, Master of Logic Thesis (MoL) Series MoL-2001-11.

[101] Gabelaia, D., R. Kontchakov, A. Kurucz, F. Wolter and M. Zakharyaschev, *Combining spatial and temporal logics: expressiveness vs. complexity*, Journal of Artificial Intelligence Research **23** (2005), pp. 167–243.

[102] Gabelaia, D., A. Kurucz, F. Wolter and M. Zakharyaschev, *Non-primitive recursive decidability of products of modal logics with expanding domains* (2006), to appear in Annals of Pure and Applied Logic.

[103] Gerson, M., *An extension of* S4 *complete for the neighbourhood semantics but incomplete for the relational semantics*, Studia Logica **34** (1975), pp. 333–342.

[104] Gerson, M., *The inadequacy of the neighbourhood semantics for modal logic*, The Journal of Symbolic Logic **40** (1975), pp. 141–147.

[105] Gettier, E., *Is Justified True Belief Knowledge?*, Analysis **23** (1963), pp. 121–123.

[106] Ghilardi, S., *Unification in intuitionistic logic*, The Journal of Symbolic Logic **64** (1999), pp. 859-880.

[107] Ghilardi, S., *Best solving modal equations*, Annals of Pure and Applied Logic **102** (2000), pp. 183-198.

[108] Girard, J., Y. Lafont and P. Taylor, "Proofs and Types," Cambridge University Press, 1989.

[109] Gödel, K., *Eine Interpretation des intuitionistischen Aussagenkalkuls*, Ergebnisse Math. Kolloq. **4** (1933), pp. 39–40, English translation in: S. Feferman et al., editors, *Kurt Gödel Collected Works. Volume I*, Oxford University Press, Oxford, Clarendon Press, New York, 1986 pp. 301–303.

[110] Gödel, K., *Vortrag bei Zilsel, 1938*, in: S. Feferman et al., editors, *Kurt Gödel Collected Works. Volume III*, Oxford University Press, 1995 pp. 86–113.

[111] Goldblatt, R., *Arithmetical necessity, provability and intuitionistic logic*, Theoria **44** (1978), pp. 38–46.

[112] Goldblatt, R., *Diodorean modality in Minkowski spacetime*, Studia Logica **39** (1980), pp. 219–237.

[113] Goldblatt, R., *Grothendieck topology as geometric modality*, Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik **27** (1981), pp. 495–529.

[114] Goldblatt, R., "Mathematics of Modality," CSLI Lecture Notes **43**, CSLI Publications, Stanford, 1993.

[115] Goldblatt, R., *Mathematical modal logic: A view of its evolution*, Journal of Applied Logic **1** (2003), pp. 309–392.

[116] Goldman, A., *A causal theory of knowing*, The Journal of Philosophy **64** (1967), pp. 335–372.

[117] Goranko, V. and S. Passy, *Using the universal modality: gains and questions*, Journal of Logic and Computation **2** (1992), pp. 5–30.

[118] Goranko, V. and D. Vakarelov, *Universal algebra and modal logic*, in: M. Zakharyaschev, K. Segerberg, M. de Rijke and H. Wansing, editors, *Advances in Modal Logic. Volume 2*, CSLI Lecture Notes **119**, CSLI Publications, Stanford, 2001 pp. 265–292.

[119] Goris, E., *Logic of proofs for bounded arithmetic*, in: *Computer Science - Theory and Applications*, Lecture Notes in Computer Science **3967** (2006), pp. 191–201.

[120] Goris, E. and J. Joosten, *Modal matters in interpretability logics*, Technical report, Utrecht University. Institute of Philosophy (2004), Logic Group preprint series; 226.

[121] Grzegorczyk, A., *Undecidability of some topological theories*, Fundamenta Mathematicae **38** (1951), pp. 137–152.

[122] Grzegorczyk, A., *Some relational systems and the associated topological spaces*, Fundamenta Mathematicae **60** (1967), pp. 223–231.

[123] Guaspari, D. and R. Solovay, *Rosser sentences*, Annals of Mathematical Logic **16** (1979), pp. 81–99.

[124] Hájek, P. and F. Montagna, *The logic of* Π_1*-conservativity*, Archive for Mathematical Logic **30** (1990), pp. 113–123.

[125] Hájek, P. and F. Montagna, *The logic of* Π_1*-conservativity continued*, Archive for Mathematical Logic **32** (1992), pp. 57–63.

[126] Hájek, P. and P. Pudlák, "Metamathematics of First-Order Arithmetic," Springer-Verlag, Berlin, Heidelberg, New York, 1993.

[127] Hamkins, J., *A simple maximality principle*, The Journal of Symbolic Logic **68** (2003), pp. 529–550.

[128] Hamkins, J. and B. Löwe, *The modal logic of forcing* (2005), manuscript.

[129] Hendricks, V., *Active agents*, JoLLI - The Journal of Language, Logic and Information (2003), pp. 469–495.

[130] Heyting, A., *Die intuitionistische grundlegung der mathematik*, Erkenntnis **2** (1931), pp. 106–115.

[131] Heyting, A., "Mathematische Grundlagenforschung. Intuitionismus. Beweistheorie," Springer, Berlin, 1934.

[132] Heyting, A., "Intuitionism: An Introduction," North-Holland, Amsterdam, 1956.

[133] Hilbert, D. and P. Bernays, "Grundlagen der Mathematik, Vols. I and II, 2d ed." Springer-Verlag, Berlin, 1968.

[134] Hintikka, J, *Impossible possible worlds vindicated*, Journal of Philosophical Logic **4** (1975), pp. 475–484.

[135] Iemhoff, R., *A modal analysis of some principles of the provability logic of Heyting arithmetic*, in: M. Zakharyaschev, K. Segerberg, M. de Rijke and H. Wansing, editors, *Advances in Modal Logic. Volume 2*, CSLI Lecture Notes **119**, CSLI Publications, Stanford, 2001 pp. 301–336.

[136] Iemhoff, R., *On the admissible rules of intuitionistic propositional logic*, The Journal of Symbolic Logic **66** (2001), pp. 281–294.

[137] Iemhoff, R., "Provability logic and admissible rules," Ph.D. thesis, University of Amsterdam, Amsterdam (2001).

[138] Iemhoff, R., *Towards a proof system for admissibility*, in: M. Baaz and A. Makowsky, editors, *Computer Science Logic '03*, Lecture Notes in Computer Science **2803**, Springer, 2003 pp. 255-270.

[139] Iemhoff, R., *Intermediate logics and Visser's rules*, Notre Dame Journal of Formal Logic **46** (2005), pp. 65-81.

[140] Ignatiev, K., *Partial conservativity and modal logics*, ITLI Prepublication Series X–91–04, University of Amsterdam (1991).

[141] Ignatiev, K., *On strong provability predicates and the associated modal logics*, The Journal of Symbolic Logic **58** (1993), pp. 249–290.

[142] Ignatiev, K., *The provability logic for Σ_1-interpolability*, Annals of Pure and Applied Logic **64** (1993), pp. 1–25.

[143] Japaridze, G., "The modal logical means of investigation of provability," Ph.D. thesis, Moscow State University (1986), in Russian.

[144] Japaridze, G., *The polymodal logic of provability*, in: *Intensional Logics and Logical Structure of Theories: Material from the fourth Soviet–Finnish Symposium on Logic, Telavi, May 20–24, 1985*, Metsniereba, Tbilisi, 1988 pp. 16–48, in Russian.

[145] Japaridze, G., *Introduction to computability logic*, Annals of Pure and Applied Logic **123** (2003), pp. 1–99.

[146] Japaridze, G., *From truth to computability I*, Theoretical Computer Science **357** (2006), pp. 100–135.

[147] Japaridze., G. and D. de Jongh, *The Logic of Provability*, in: S. Buss, editor, *Handbook of Proof Theory*, Elsevier, 1998 pp. 475–546.

[148] Joosten, J., "Interpretability formalized," Ph.D. thesis, University of Utrecht (2004).

[149] Kleene, S., "Introduction to Metamathematics," Van Norstrand, 1952.

[150] Kolmogoroff, A., *Zur Deutung der intuitionistischen logik*, Mathematische Zeitschrift **35** (1932), pp. 58–65, in German. English translation in: V.M. Tikhomirov, editor, *Selected works of A.N. Kolmogorov. Volume I: Mathematics and Mechanics*, Kluwer, Dordrecht, 1991 pp. 151–158.

[151] Konev, B., R. Kontchakov, F. Wolter and M. Zakharyaschev, *On dynamic topological and metric logics*, in: R. Schmidt, I. Pratt-Hartmann, M. Reynolds, H. Wansing, editors, *Proceedings of AiML 2004*, Manchester, U.K., 2004 pp. 182–196.

[152] Konev, B., R. Kontchakov, F. Wolter and M. Zakharyaschev, *Dynamic topological logics over spaces with continuous functions*, 2006. Manuscript. Available at http://www.dcs.bbk.ac.uk/∼michael/

[153] Kracht, M., "Tools and Techniques in Modal Logic," Studies in Logic and the Foundations of Mathematics **142**, Elsevier, Amsterdam, 1999.

[154] Kremer, P., *Temporal logic over* S4*: an axiomatizable fragment of dynamic topological logic*, The Bulletin of Symbolic Logic **3** (1997), pp. 375–376.

[155] Kremer, P. and G. Mints, *Dynamic topological logic*, The Bulletin of Symbolic Logic **3** (1997), pp. 371–372.

[156] Kremer, P. and G. Mints, *Dynamic topological logic*, Annals of Pure and Applied Logic **131** (2005), pp. 133–158.

[157] Kremer, P., G. Mints and V. Rybakov, *Axiomatizing the next-interior fragment of dynamic topological logic*, The Bulletin of Symbolic Logic **3** (1997), pp. 376–377.

[158] Kripke, S., *Semantical considerations on modal logic*, Acta Philosophica Fennica **16** (1963), pp. 83–94.

[159] Krupski, N., *On the complexity of the reflected logic of proofs*, Theoretical Computer Science **357** (2006), pp. 136–142.

[160] Krupski, V., *Operational logic of proofs with functionality condition on proof predicate*, in: S. Adian and A. Nerode, editors, *Logical Foundations of Computer Science '97, Yaroslavl*, Lecture Notes in Computer Science **1234**, Springer, 1997 pp. 167–177.

[161] Krupski, V., *The single-conclusion proof logic and inference rules specification*, Annals of Pure and Applied Logic **113** (2001), pp. 181–206.

[162] Kunen, K., "Set Theory: An Introduction to Independence Proofs," North Holland, 1980.

[163] Kuratowski, C., *Sur l'operation a de l'analysis situs*, Fundamenta Mathematicae **3** (1922), pp. 181–199.

[164] Kuratowski, K., "Topology. Volume 1," Academic Press, New York, 1966.

[165] Kurucz, A., F. Wolter and M. Zakharyaschev, *Modal logics for metric spaces: open problems*, in: S. Artemov, H. Barringer, A. d'Avila Garcez, L. Lamb and J. Woods, editors, *We Will Show Them: Essays in Honour of Dov Gabbay. Volume 2*, College Publications, London, 2005 pp. 193–208.

[166] Kutz, O., H. Sturm, N.-Y. Suzuki, F. Wolter and M. Zakharyaschev, *Logics of metric spaces*, ACM Transactions in Computational Logic **4** (2003), pp. 260–294.

[167] Kuznets, R., *Logic of Proofs as a measure of Hilbert-style complexity*, The Bulletin of Symbolic Logic **12** (2006), pp. 355.

[168] Kuznets, R., *On the complexity of explicit modal logics*, in: *Computer Science Logic 2000*, Lecture Notes in Computer Science **1862**, Springer-Verlag, 2000 pp. 371–383.

[169] Kuznetsov, A. and A. Muravitsky, *The logic of provability*, in: *Abstracts of the 4th All-Union Conference on Mathematical Logic*, Kishinev, 1976 p. 73, in Russian.

[170] Kuznetsov, A. and A. Muravitsky, *Magari algebras*, in: *Fourteenth All-Union Algebra Conference, Abstracts, Part 2: Rings, Algebraic Structures*, 1977 pp. 105–106, in Russian.

[171] Kuznetsov, A. and A. Muravitsky, *On superintuitionistic logics as fragments of proof logic*, Studia Logica **XLV** (1986), pp. 76–99.

[172] Lavendhomme, R., T. Lucas and G. Reyes, *Formal systems for topos-theoretic modalities*, Bull. Soc. Math. Belg. Sr. A **41** (1989), pp. 333–372.

[173] Lawvere, F., *Quantifiers and sheaves*, in: *Actes du Congré International des Mathématiciens. Tome 1*, 1970 pp. 329–334.

[174] Lehrer, K. and T. Paxson, *Knowledge: undefeated justified true belief*, The Journal of Philosophy **66** (1969), pp. 1–22.

[175] Leibman, G., "Consistency strengths of Maximality principles," Ph.D. thesis, The Graduate Center of the City University of New York (2004).

[176] Lemmon, E., *New foundations for Lewis's modal systems*, The Journal of Symbolic Logic **22** (1957), pp. 176–186.

[177] Lenzen, W., *Knowledge, belief and subjective probability*, in: V. Hendricks and S. Pedersen, editors, *Knowledge Contributors*, Kluwer, 2003 .

[178] Lewis, D., *Elusive knowledge*, Australian Journal of Philosophy **7** (1996), pp. 549–567.

[179] Lindström, P., *Provability logic – a short introduction*, Theoria **62** (1996), pp. 19–61.

[180] Löb, M., *Solution of a problem of Leon Henkin*, The Journal of Symbolic Logic **20** (1955), pp. 115–118.

[181] Löwe, B. and D. Sarenac, *Cardinal spaces and topological representations of bimodal logics*, to appear.

[182] Macintyre, A. and H. Simmons, *Gödel's diagonalization technique and related properties of theories*, Colloquium Mathematicum **28** (1973), pp. 165-180.

[183] Magari, R., *The diagonalizable algebras (the algebraization of the theories which express Theor.:II)*, Bollettino della Unione Matematica Italiana, Serie 4 **12** (1975), suppl. fasc. 3, pp. 117–125.

[184] Magari, R., *Representation and duality theory for diagonalizable algebras (the algebraization of theories which express Theor.:IV)*, Studia Logica **34** (1975), pp. 305–313.

[185] Makkai, M. and G. Reyes, *Completeness results for intuitionistic and modal logic in a categorical setting*, Annals of Pure and Applied Logic **72** (1995), pp. 25–101.

[186] Mazurkiewicz, S. and W. Sierpinski, *Contribution a la topologie des ensambles donomrables*, Fundamenta Mathematicae **1** (1920), pp. 17–27.

[187] McKinsey, J. and A. Tarski, *The algebra of topology*, Annals of Mathematics **45** (1944), pp. 141–191.

[188] McKinsey, J. and A. Tarski, *Some theorems about the sentential calculi of Lewis and Heyting*, The Journal of Symbolic Logic **13** (1948), pp. 1–15.

[189] Meyer, J.-J. Ch. and W. van der Hoek, "Epistemic Logic for AI and Computer Science," Cambridge University Press, 1995.

[190] Milnikel, R., *Derivability in certain subsystems of the Logic of Proofs is Π_2^p-complete*, Annals of Pure and Applied Logic (2006), to appear.

[191] Mints, G., *Lewis' systems and system* T *(a survey 1965-1973).*, in: R. Feys. *Modal Logic (Russian translation)*, Nauka, Moscow, 1974 pp. 422–509, in Russian, English translation in G. Mints, *Selected papers in proof theory*, Bibliopolis, Napoli, 1992.

[192] Mints, G., *A completeness proof for propositional* S4 *in Cantor space*, in: E. Orlowska, editor, *Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa*, Physica-Verlag, Heidelberg, 1998 .

[193] Mkrtychev, A., *Models for the Logic of Proofs*, in: S. Adian and A. Nerode, editors, *Logical Foundations of Computer Science '97, Yaroslavl*, Lecture Notes in Computer Science **1234**, Springer, 1997 pp. 266–275.

[194] Montagna, F., *On the diagonalizable algebra of Peano arithmetic*, Bollettino della Unione Matematica Italiana, B (5) **16** (1979), pp. 795–812.

[195] Montagna, F., *Undecidability of the first order theory of diagonalizable algebras*, Studia Logica **39** (1980), pp. 347–354.

[196] Montagna, F., *The predicate modal logic of provability*, Notre Dame Journal of Formal Logic **25** (1987), pp. 179–189.

[197] Montague, R., *Syntactical treatments of modality with corollaries on reflection principles and finite axiomatizability*, Acta Philosophica Fennica **16** (1963), pp. 153–168.

[198] Moses, Y., *Resource-bounded knowledge*, in: M. Vardi, editor, *Theoretical Aspects of Reasoning about Knowledge*, 1988 pp. 261–276.

[199] Myhill, J., *Some remarks on the notion of proof*, Journal of Philosophy **57** (1960), pp. 461–471.

[200] Myhill, J., *Intensional set theory*, in: S. Shapiro, editor, *Intensional Mathematics*, North-Holland, 1985 pp. 47–61.

[201] Nogina, E., *On logic of proofs and provability*, The Bulletin of Symbolic Logic **12** (2006), pp. 356.

[202] Nogina, E., *Logic of proofs with the strong provability operator*, Technical Report ILLC Prepublication Series ML-94-10, Institute for Logic, Language and Computation, University of Amsterdam (1994).

[203] Nogina, E., *Grzegorczyk logic with arithmetical proof operators*, Fundamental and Applied Mathematics **2** (1996), pp. 483–499, in Russian.

[204] Nozick, R., "Philosophical Explanations," Harvard University Press, 1981.

[205] Orlov, I., *The calculus of compatibility of propositions*, Matematicheskii Sbornik **35** (1928), pp. 263–286, in Russian.

[206] Parikh, R., *Knowledge and the problem of logical omniscience*, in: Z. Ras and M. Zemankova, editors, ISMIS-87 *(International Symposium on Methodolody for Intellectual Systems)*, 1987 pp. 432–439.

[207] Parikh, R., *Logical omniscience*, in: D. Leivant, editor, *Logic and Computational Complexity, International Workshop LCC '94, Indianapolis, Indiana, USA, 13-16 October 1994*, Lecture Notes in Computer Science **960**, Springer, 1995 pp. 22–29.

[208] Prior, A., "Time and Modality," Clarendon Press, Oxford, 1957.

[209] Prior, A., "Past, Present and Future," Oxford University Press, 1967.

[210] Prior, A., "Papers on Time and Tense," Clarendon Press, Oxford, 1969.

[211] Randell, D., Z. Cui and A. Cohn, *A spatial logic based on regions and connection*, in: B. Nebel, C. Rich, and W. Swartout, editors, *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, Morgan Kaufmann, 1992 pp. 165–176.

[212] Renz, J. and B. Nebel, *Spatial reasoning with topological information*, in: C. Freksa, C. Habel and K. Wender, editors, *Spatial Congnition – An Interdisciplinary Approach to Representation and Processing of Spatial Knowledge*, Lecture Notes in Computer Science **1404**, Springer, 1998 pp. 351–372.

[213] Reyes, G., *A topos-theoretic approach to reference and modality*, Notre Dame Journal of Formal Logic **32** (1991), pp. 359–391.

[214] Reyes, G. and H. Zolfaghari, *Topos-theoretic approaches to modality*, in: A. Carboni, C. Pedicchio and G. Rosolini, editors, *Conference on Category theory '90*, Lecture Notes in Mathematics **1488**, Springer, 1991 pp. 359–378.

[215] Reyes, G. and H. Zolfaghari, *Bi-Heyting algebras, toposes and modalities*, Journal of Philosophical Logic **25** (1996), pp. 25–43.

[216] Riesz, F., *Stetigkeitsbegriff und abstrakte mengenlehre*, in: *Atti del IV Congr. Internat. d. Mat., v. II* (1909).

[217] Robinson, A., *Metamathematical problems*, The Journal of Symbolic Logic **38** (1973), pp. 159–171.

[218] Rosser, J., *Extensions of some theorems of Gödel and Church*, The Journal of Symbolic Logic **1** (1936), pp. 87–91.

[219] Sarenac, D., "Modal logic and topological products," Ph.D. thesis, Stanford University (2005).

[220] Scott, D., *Advice on modal logic*, in: K. Lampert, editor, *Philosophical Problems in Logic*, Reidel, Dordrecht, Netherlands, 1970 pp. 143–173.

[221] Shapiro, S., *Epistemic and intuitionistic arithmetic*, in: S. Shapiro, editor, *Intensional Mathematics*, North-Holland, 1985 pp. 11–46.

[222] Shapiro, S., *Intensional mathematics and constructive mathematics*, in: S. Shapiro, editor, *Intensional Mathematics*, North-Holland, 1985 pp. 1–10.

[223] Shapirovsky, I. and V. Shehtman, *Chronological future modality in Minkowski spacetime*, in: P. Balbiani, N.-Y. Suzuki, F. Wolter and M. Zakharyaschev, editors, *Advances in Modal Logic. Volume 4*, King's College Publications, London, 2003 pp. 437–459.

[224] Shavrukov, V., *The logic of relative interpretability over Peano arithmetic*, Preprint, Steklov Mathematical Institute, Moscow (1988), in Russian.

[225] Shavrukov, V., *On Rosser's provability predicate*, Zeitschrift f. math. Logik und Grundlagen d. Math. **37** (1991), pp. 317–330.

[226] Shavrukov, V., *A smart child of Peano's*, Notre Dame Journal of Formal Logic **35** (1994), pp. 161–185.

[227] Shavrukov, V., *Isomorphisms of diagonalizable algebras*, Theoria **63** (1997), pp. 210–221.

[228] Shehtman, V., *Topological models of propositional logics*, in: *Semiotika i informatika*, 15, VINITI, Moscow, 1980 pp. 74–98, in Russian.

[229] Shehtman, V., *Modal logic of domains on the real plane*, Studia Logica **42** (1983), pp. 63–80.

[230] Shehtman, V., *Derived sets in Euclidean spaces and modal logic*, Technical Report ITLI Prepublication Series, X-90-05, University of Amsterdam (1990).

[231] Shehtman, V., *"Everywhere" and "here,"* Journal of Applied Non-Classical Logics **9** (1999), pp. 369–380.

[232] Shehtman, V., *Derivational modal logics* (2005), submitted to Moscow Mathematical Journal.

[233] Shehtman, V., *On neighbourhood semantics thirty years later* (2005), forthcoming.

[234] Sheremet, M,, D. Tishkovsky, F. Wolter and M. Zakharyaschev, *Comparative similarity, tree automata, and Diophantine equations*, in: G. Sutcliffe and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning: 12th International Conference, LPAR 2005, Montego Bay, Jamaica, December 2-6, 2005. Proceedings*, Lecture Notes in Artificial Intelligence **3835**, Springer, 2005 pp. 651–665.

[235] Sidon, T., *Provability logic with operations on proofs*, in: S. Adian and A. Nerode, editors, *Logical Foundations of Computer Science '97, Yaroslavl*, Lecture Notes in Computer Science **1234**, Springer, 1997 pp. 342–353.

[236] Slavnov, S., *Two counterexamples in the logic of dynamic topological systems*, Technical Report TR-2003015, CUNY Ph.D. Program in Computer Science (2003).

[237] Slavnov, S., *On completeness of dynamic topological logic*, Moscow Mathematical Journal **5** (2005), pp. 477–492.

[238] Smiley, T., *The logical basis of ethics*, Acta Philosophica Fennica **16** (1963), pp. 237–246.

[239] Smoryński, C., *Applications of Kripke models*, in: A. Troelstra, editor, *Mathematical Investigations of Intuitionistic Arithmetic and Analysis*, Springer Verlag, 1973 pp. 324-391.

[240] Smoryński, C., *The incompleteness theorems*, in: J. Barwise, editor, *Handbook of Mathematical Logic*, North Holland, Amsterdam, 1977 pp. 821–865.

[241] Smoryński, C., "Self-Reference and Modal Logic," Springer-Verlag, Berlin, 1985.

[242] Solovay, R., *Provability interpretations of modal logic*, Israel Journal of Mathematics **25** (1976), pp. 287–304.

[243] Švejdar, V., *Modal analysis of generalized Rosser sentences*, The Journal of Symbolic Logic **48** (1983), pp. 986–999.

[244] Takeuti, G., "Proof Theory," Elsevier Science Ltd; 2nd Rev edition, 1987.

[245] Tarski, A., A. Mostowski and R. Robinson, "Undecidable Theories," North-Holland, 1953.

[246] Troelstra, A. and D. van Dalen, "Constructivism in Mathematics, vols 1, 2," North–Holland, Amsterdam, 1988.

[247] van Benthem, J., *Reflections on epistemic logic*, Logique & Analyse **133-134** (1993), pp. 5–14.

[248] van Benthem, J., *Modal frame correspondence generalized*, Technical Report PP-2005-08, Institute for Logic, Language, and Computation, Amsterdam (2005).

[249] van Benthem, J. and J. Bergstra, *Logics of transition systems*, Journal of Logic, Language, and Information **3** (1995), pp. 247–284.

[250] van Benthem, J., G. Bezhanishvili and M. Gehrke, *Euclidean hierarchy in modal logic*, Studia Logica **75** (2003), pp. 327–344.

[251] van Benthem, J., G. Bezhanishvili, B. ten Cate and D. Sarenac, *Multimodal logics for products of topologies* (2006), to appear in Studia Logica.

[252] Vardanyan, V., *Arithmetic comlexity of predicate logics of provability and their fragments*, *Soviet Mathematics Doklady* **33** (1986), pp. 569–572.

[253] Visser, A., "Aspects of Diagonalization and Provability," Ph.D. thesis, University of Utrecht, Utrecht, The Netherlands (1981).

[254] Visser, A., *The provability logics of recursively enumerable theories extending Peano Arithmetic at arbitrary theories extending Peano Arithmetic*, Journal of Philosophic Logic **13** (1984), pp. 97–113.

[255] Visser, A., *Evaluation, provably deductive equivalence in Heyting Arithmetic of substitution instances of propositional formulas*, Logic Group Preprint Series 4, Department of Philosophy, University of Utrecht (1985).

[256] Visser, A., *Peano's smart children. A provability logical study of systems with built-in consistency*, Notre Dame Journal of Formal Logic **30** (1989), pp. 161–196.

[257] Visser, A., *Interpretability logic*, in: P. Petkov, editor, *Mathematical Logic*, Plenum Press, New York, 1990 pp. 175–208.

[258] Visser, A., *Propositional combinations of Σ_1-sentences in Heyting's Arithmetic*, Logic Group Preprint Series 117, Department of Philosophy, University of Utrecht (1994).

[259] Visser, A., *An overview of interpretability logic*, in: M. Kracht, M. de Rijke and H. Wansing, editors, *Advances in Modal Logic. Volume 1*, CSLI Publications, Stanford University, 1998 pp. 307–360.

[260] Visser, A., *Rules and arithmetics*, Notre Dame Journal of Formal Logic **40** (1999), pp. 116–140.

[261] Visser, A., *Substitutions of Σ^0_1-sentences: Explorations between intuitionistic propositional logic and intuitionistic arithmetic*, Annals of Pure and Applied Logic **114** (2002), pp. 227–271.

[262] Visser, A., *Löb's logic meets the μ-calculus*, in: A. Middeldorp, V. van Oostrom, F. van Raamsdonk and R. de Vrijer, editors, *Processes, Terms and Cycles: Steps on the Road to Infinity. Essays Dedicated to Jan Willem Klop on the Occasion of his 60th Birthday*, Lecture Notes in Computer Science **3838** (2005), pp. 14–25.

[263] von Neumann, J., *A letter to Gödel on January 12, 1931*, in: S. Feferman, J. Dawson, W. Goldfarb, C. Parsons and W. Sieg, editors, *Kurt Gödel Collected Works. Volume V*, Oxford University Press, 2003 pp. 341–345.

[264] von Wright, G., *''And next,''* Acta Philosophica Fennica **18** (1965), pp. 293–304.

[265] von Wright, G., *"Always,"* Theoria **34** (1968), pp. 208–221.

[266] Wolter, F. and M. Zakharyaschev, *Spatial reasoning in RCC-8 with Boolean region terms*, in: *Proceedings of the 14th European Conference in Artificial Intelligence (ECAI 2000)*, 2000 pp. 244–248.

[267] Wolter, F. and M. Zakharyaschev, *Reasoning about distances*, in: *18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*, 2003 pp. 1275–1280.

[268] Wolter, F. and M. Zakharyaschev, *A logic for metric and topology*, The Journal of Symbolic Logic **70** (2005), pp. 795–828.

[269] Yavorskaya (Sidon), T., *Logic of proofs and provability*, Annals of Pure and Applied Logic **113** (2001), pp. 345–372.

[270] Yavorsky, R., *On the logic of the standard proof predicate*, in: *Computer Science Logic 2000*, Lecture Notes in Computer Science **1862**, Springer, 2000 pp. 527–541.

[271] Yavorsky, R., *Provability logics with quantifiers on proofs*, Annals of Pure and Applied Logic **113** (2001), pp. 373–387.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

17

AUTOMATA-THEORETIC TECHNIQUES FOR TEMPORAL REASONING

Moshe Y. Vardi

1	Introduction	971
2	Automata Theory	973
2.1	Words and Trees	973
2.2	Nondeterministic Automata on Infinite Words	974
2.3	Alternating Automata on Infinite Words	974
2.4	Nondeterministic Automata on Infinite Trees	976
2.5	Alternating Automata on Infinite Trees	976
3	Temporal Logics and Alternating Automata	979
3.1	Linear Temporal Logic	979
3.2	Branching Temporal Logic	980
4	Model Checking	982
4.1	Linear Temporal Logic	983
4.2	Branching Temporal Logic	983
5	Validity Checking	985
5.1	Linear Temporal Logic	985
5.2	Branching Temporal Logic	985

1 INTRODUCTION

This chapter describes an automata-theoretic approach to temporal reasoning. The basic idea underlying this approach is that for any temporal formula we can construct a finite-state automaton that accepts the computations that satisfy the formula. For linear temporal logics the automaton runs on infinite words while for branching temporal logics the automaton runs on infinite trees. The simple combinatorial structures that emerge from the automata-theoretic approach decouple the logical and algorithmic components of temporal reasoning and yield clear and asymptotically optimal algorithms.

Temporal logics, which are modal logics geared towards the description of the temporal ordering of events, have been adopted as a powerful tool for specifying behavior concurrent programs and for verifying that such programs meet their specifications [47, 57]. One of the most significant developments in this area is the discovery of algorithmic methods for verifying temporal logic properties of *finite-state* programs [15, 44, 60, 81].

(A state of a program is a complete description of its status, including the assignment of values to variables, the value of the program counter, which points to the instruction currently being executed, and the like. Finite-state programs have finitely many possible states, which means that the variables range over finite domains and recursion, if allowed, is bounded in depth.) This derives its significance from the fact that many synchronization, coordination and communication protocols can be modeled as finite-state programs [46, 63]. Finite-state programs can be modeled by transition systems where each state has a bounded description, and hence can be characterized by a fixed number of Boolean atomic propositions. This means that a finite-state program can be viewed as a finite *propositional Kripke structure* and that its properties can be specified using *propositional* temporal logic. Thus, to verify the correctness of the program with respect to a desired behavior, one only has to check that the program, modeled as a finite Kripke structure, is a model of (satisfies) the propositional temporal logic formula that specifies that behavior. Hence the name *model checking* for the verification methods derived from this viewpoint. An extensive survey of model checking can be found in [16, 37]. Model checking is an *algorithmic* approach to program verification. It is different from the *deductive approach*, in which in which a person, perhaps aided by computers, use deductive techniques to prove that a program satisfy its specification [48].

We distinguish between two types of temporal logics: linear and branching (see [43] and general discussion of temporal structures in Chapter 11 of this handbook). In linear temporal logics, each moment in time has a unique possible future, while in branching temporal logics, each moment in time may split into several possible futures. (For an extensive discussion of various temporal logics, see [22].) For both types of temporal logics, a close and fruitful connection with the theory of automata on infinite structures has been developed. The basic idea is to associate with each temporal logic formula a finite automaton on infinite structures that accepts the computations that satisfy the formula. For linear temporal logic the structures are infinite words [66, 45, 68, 83], while for branching temporal logic the structures are infinite trees [30, 69, 21, 26, 82]. This enables the reduction of temporal logic decision problems, such as satisfiability and model checking, to known automata-theoretic problems, such as nonemptiness, yielding clean and asymptotically optimal algorithms. This reduction is the subject matter of this chapter. (See also Chapter 11 for a general discussion of temporal reasoning and Chapters 4 and 7 for discussions of modal reasoning.)

Initially, the translations in the literature from temporal logic formulas to automata used *nondeterministic* automata (cf. [34, 82, 83]). These translations have two disadvantages. First, the translation itself is rather nontrivial; For example, in [82, 83] the translations go through a series of ad-hoc intermediate representations in an attempt to simplify the translation. Second, for both linear and branching temporal logics there is an exponential blow-up involved in going from formulas to automata. This suggests that any algorithm that uses these translations as one of its steps is going to be an exponential-time algorithm. Thus, the automata-theoretic approach did not seem to be applicable to branching-time model checking, which in many cases can be done in linear execution time [15, 17, 60].

In the mid 1990s, it was shown that if one uses *alternating* automata rather than *nondeterministic* automata, then these problems can be addressed [42, 74]. Alternating automata generalize the standard notion of nondeterministic automata by allowing several successor states to go down along the same word or the same branch of the tree. In

this chapter we argue that *alternating automata* offer the key to a comprehensive and satisfactory automata-theoretic framework for temporal logics. We demonstrate this claim by showing how alternating automata can be used to derive model-checking and satisfiability algorithms for both linear and branching temporal logics. The key observation is that while the translation from temporal logic formulas to nondeterministic automata is exponential [82, 83], the translation to alternating automata is linear [27, 42, 53, 74]. Thus, the advantage of alternating automata is that they enable one to decouple the logic from the combinatorics. The translations from formulas to automata handle the logic, and the algorithms that handle the automata handles the combinatorics.

Historical Note: The connection between logic and automata goes back to work in the early 1960s [6, 20, 73] on monadic second-order logic and automata over finite words. This was extended in [7] to infinite words, in [19, 72] to finite trees, and in [61] to infinite trees. As temporal logics can be expressed in first-order or monadic second-order logic [38, 35], the connection between monadic second-order logic and automata yields a connection between temporal logics and automata. Developing decision procedures that go via monadic second-order logic was a standard approach in the 1970s, see [33]. A direct translation to automata was proposed first in [70] in the context of propositional dynamic logic. A direct translation from temporal logic to automata was first given in [85] (see also [83] for linear time and in [80] for branching time). The translation to alternating automata was first proposed in [53] and pursued further in [42, 74, 75].

2 AUTOMATA THEORY

2.1 Words and Trees

We are given a finite nonempty alphabet Σ. A *finite word* is an element of Σ^*, i.e., a finite sequence $a_0, \ldots, a_n$ of symbols from Σ. An *infinite word* is an element of Σ^ω, i.e., an infinite sequence $a_0, a_1, \ldots$ of symbols from Σ.

A *tree* is a (finite or infinite) connected directed graph, with one node designated as the *root* and denoted by ε, and in which every non-root node has a unique parent (s is the *parent* of t and t is a *child* of s if there is an edge from s to t) and the root ε has no parent. The *arity* of a node x in a tree τ, denoted $arity(x)$, is the number of children of x in τ. The *level* of a node x, denoted $|x|$, is its distance from the root; in particular, $|\varepsilon| = 0$. Let N denote the set of positive integers. A *tree* τ *over* N is a subset of N^*, such that if $x \cdot i \in \tau$, where $x \in N^*$ and $i \in N$, then $x \in \tau$, there is an edge from x to $x \cdot i$, and if $i > 1$ then also $x \cdot (i-1) \in \tau$. By definition, the empty sequence ε is the root of such a tree. Let $\mathcal{D} \subseteq N$. We say that a tree τ is a $\mathcal{D}$*-tree* if τ is a tree over N and $arity(x) \in \mathcal{D}$ for all $x \in \tau$. A tree is called *leafless* if every node has at least one child.

A *branch* $\beta = x_0, x_1, \ldots$ of a tree is a maximal sequence of nodes such that x_0 is the root and x_i is the parent of x_{i+1} for all $i > 0$. Note that β can be finite or infinite; if it is finite, then the last node of the branch has no children. A Σ*-labeled tree*, for a finite alphabet Σ, is a pair $(\tau, \mathcal{T})$, where τ is a tree and $\mathcal{T}$ is a mapping $\mathcal{T} : nodes(\tau) \rightarrow \Sigma$ that assigns to every node a label. We often refer to $\mathcal{T}$ as the labeled tree, leaving its domain implicit. A branch $\beta = x_0, x_1, \ldots$ of $\mathcal{T}$ defines a word $\mathcal{T}(\beta) = \mathcal{T}(x_0), \mathcal{T}(x_1), \ldots$ consisting of the sequence of labels along the branch.

2.2 Nondeterministic Automata on Infinite Words

A *nondeterministic Büchi automaton* A on words is a tuple $(\Sigma, S, S^0, \rho, F)$, where Σ is a finite nonempty *alphabet*, S is a finite nonempty set of *states*, $S^0 \subseteq S$ is the set of *initial* states, $F \subseteq S$ is the set of *accepting* states, and $\rho : S \times \Sigma \to 2^S$ is a *transition function.* Intuitively, $\rho(s, a)$ is the set of states that A can move into when it is in state s and it reads the symbol a. Note that the automaton may be nondeterministic, since it may have many initial states and the transition function may specify many possible transitions for each state and symbol.

A run r of A on an infinite word $w = a_0, a_1, \ldots$ over Σ is a sequence $s_0, s_1, \ldots$, where $s_0 \in S^0$ and $s_{i+1} \in \rho(s_i, a_i)$, for all $i \geq 0$. We define $\lim(r)$ to be the set $\{s \,|\, s = s_i \text{ for infinitely many } i\text{'s}\}$, i.e., the set of states that occur in r infinitely often. Since S is finite, $\lim(r)$ is necessarily nonempty. The run r is *accepting* if there is some accepting state that repeats in r infinitely often, i.e., $\lim(r) \cap F \neq \emptyset$. The infinite word w is *accepted* by A if there is an accepting run of A on w. The set of infinite words accepted by A is denoted $L_\omega(A)$.

An important feature of nondeterministic Büchi automata is their closure under intersection.

PROPOSITION 1. [12] *Let A_1 and A_2 be nondeterministic Büchi automata with n_1 and n_2 states, respectively. Then there is a Büchi automaton A with $O(n_1 n_2)$ states such that $L_\omega(A) = L_\omega(A_1) \cap L_\omega(A_2)$.*

One of the most fundamental algorithmic issues in automata theory is testing whether a given automaton is "interesting", i.e., whether it accepts some input. A Büchi automaton A is *nonempty* if $L_\omega(A) \neq \emptyset$. The *nonemptiness problem* for automata is to decide, given an automaton A, whether A is nonempty. It turns out that testing nonemptiness for Büchi automata is easy A accepts some word iff in the graph $G_A = (S, E_A)$, where $E_A = \{(s,t) | t \in \rho(s,a) \text{ for some } a \in \Sigma\}$, there is a path from S_0 that reaches some state $f \in F$ and then cycles back to f. This can be checked using depth-first search [71] or space-efficient search [65].

PROPOSITION 2.

1. [29, 28] *The nonemptiness problem for nondeterministic Büchi automata is decidable in linear time.*

2. [83] *The nonemptiness problem for nondeterministic Büchi automata of size n is decidable in space $O(\log^2 n)$.*

2.3 Alternating Automata on Infinite Words

Nondeterminism gives a computing device the power of existential choice. Its dual gives a computing device the power of universal choice. It is therefore natural to consider computing devices that have the power of both existential choice and universal choice. Such devices are called *alternating.* Alternation was studied in [11] in the context of Turing machines and in [5, 11] for finite automata. The alternation formalisms in [5] and [11] are different, though equivalent. We follow here the formalism of [5], which was extended in [55] to automata on infinite structures.

For a given set X, let $\mathcal{B}^+(X)$ be the set of positive Boolean formulas over X (i.e., Boolean formulas built from elements in X using $\wedge$ and $\vee$), where we also allow the formulas **true** and **false**. Let $Y \subseteq X$. We say that Y *satisfies* a formula $\theta \in \mathcal{B}^+(X)$ if the truth assignment that assigns *true* to the members of Y and assigns *false* to the members of $X - Y$ satisfes θ. For example, the sets $\{s_1, s_3\}$ and $\{s_1, s_4\}$ both satisfy the formula $(s_1 \vee s_2) \wedge (s_3 \vee s_4)$, while the set $\{s_1, s_2\}$ does not satisfy this formula.

Consider a nondeterministic automaton $A = (\Sigma, S, s^0, \rho, F)$. The transition function ρ maps a state $s \in S$ and an input symbol $a \in \Sigma$ to a set of states. Each element in this set is a possible nondeterministic choice for the automaton's next state. We can represent ρ using $\mathcal{B}^+(S)$; for example, $\rho(s, a) = \{s_1, s_2, s_3\}$ can be written as $\rho(s, a) = s_1 \vee s_2 \vee s_3$. In alternating automata, $\rho(s, a)$ can be an arbitrary formula from $\mathcal{B}^+(S)$. We can have, for instance, a transition

$$\rho(s, a) = (s_1 \wedge s_2) \vee (s_3 \wedge s_4),$$

meaning that the automaton accepts the word aw, where a is a symbol and w is a word, when it is in the state s if it accepts the word w from both s_1 and s_2 or from both s_3 and s_4. Thus, such a transition combines the features of existential choice (the disjunction in the formula) and universal choice (the conjunctions in the formula).

Formally, an *alternating Büchi automaton* is a tuple $A = (\Sigma, S, s^0, \rho, F)$, where Σ is a finite nonempty alphabet, S is a finite nonempty set of states, $s^0 \in S$ is an initial state, F is a set of accepting states, and $\rho : S \times \Sigma \rightarrow \mathcal{B}^+(S)$ is a transition function. As a convention, if $\rho(s, a)$ is not specified, we assume that it is **false**.

Because of the universal choice in alternating transitions, a run of an alternating automaton is a tree rather than a sequence. A run of A on an infinite word $w = a_0 a_1 \ldots$ is an S-labeled tree r such that $r(\varepsilon) = s^0$ and the following holds:

> if $|x| = i$, $r(x) = s$, and $\rho(s, a_i) = \theta$, then x has k children $x_1, \ldots, x_k$, for some $k \leq |S|$, and $\{r(x_1), \ldots, r(x_k)\}$ satisfies θ.

For example, if $\rho(s_0, a_0)$ is $(s_1 \vee s_2) \wedge (s_3 \vee s_4)$, then the nodes of the run tree at level 1 include the label s_1 or the label s_2 and also include the label s_3 or the label s_4. Note that the run can also have finite branches; if $|x| = i$, $r(x) = s$, and $\rho(s, a_i) = \mathbf{true}$, then x does not need to have any children. On the other hand, we cannot have $\rho(s, a_i) = \mathbf{false}$, since **false** is not satisfiable. The run is *accepting* if $\lim(r(\beta)) \cap F \neq \emptyset$ for every branch $\beta = x_0, x_1, \ldots$ of the run. that is, for every infinite branch $\beta = x_0, x_1, \ldots$, we have that $r(x_i) \in F$ for infinitely many i's. A word w is accepted by A is A has an accepting run on w.

What is the relationship between alternating Büchi automata and nondeterministic Büchi automata? It is easy to see that alternating Büchi automata generalize nondeterministic Büchi automata; nondeterministic automata correspond to alternating automata where the transitions are pure disjunctions. It turns out that they have the same expressive power (although alternating Büchi automata are more succinct than nondeterministic Büchi automata).

PROPOSITION 3. [51] *Let A be an alternating Büchi automaton with n states. Then there is a nondeterministic Büchi automaton A_{nd} with 3^n states such that $L_\omega(A_{nd}) = L_\omega(A)$.*

By combining Propositions 2 and 3 (with its exponential blowup), we can obtain a nonemptiness test for alternating Büchi automata.

PROPOSITION 4.

1. *The nonemptiness problem for alternating Büchi automata is decidable in exponential time.*

2. *The nonemptiness problem for alternating Büchi automata is decidable in quadratic space.*

2.4 Nondeterministic Automata on Infinite Trees

We now consider automata on labeled leafless $\mathcal{D}$-trees. A *nondeterministic Büchi tree automaton* A is a tuple $(\Sigma, \mathcal{D}, S, S^0, \rho, F)$. Here Σ is a finite alphabet, $\mathcal{D} \subset N$ is a finite set of arities, S is a finite set of states, $S^0 \subseteq S$ is the set of initial states, $F \subseteq S$ is a set of accepting states, and $\rho : S \times \Sigma \times \mathcal{D} \to 2^{S^*}$ is a transition function, where $\rho(s, a, k) \subseteq S^k$ for each $s \in S$, $a \in \Sigma$, and $k \in \mathcal{D}$. Thus, $\rho(s, a, k)$ is a set of k-tuples of states. Intuitively, when the automaton is in state s and it is reading a k-ary node x of a tree $\mathcal{T}$, it nondeterministically chooses a k-tuple $\langle s_1, \ldots, s_k\rangle$ in $\rho(s, \mathcal{T}(x))$, makes k copies of itself, and then moves to the node $x \cdot i$ in the state s_i for $i = 1, \ldots, k$. A *run* $r : \tau \to S$ of A on a Σ-labeled $\mathcal{D}$-tree $\mathcal{T}$ is an S-labeled $\mathcal{D}$-tree such that the root is labeled by an initial state and the transitions obey the transition function ρ; that is, $r(\varepsilon) \in S^0$, and for each node x such that $arity(x) = k$, we have $\langle r(x \cdot 1), \ldots, r(x \cdot k)\rangle \in \rho(r(x), \mathcal{T}(x), k)$. The run is *accepting* if $\lim(r(\beta)) \cap F \neq \emptyset$ for every branch $\beta = x_0, x_1, \ldots$ of τ; that is, for every branch $\beta = x_0, x_1, \ldots$, we have that $r(x_i) \in F$ for infinitely many i's. The set of trees accepted by A is denoted $T_\omega(A)$. It is easy to see that nondeterministic Büchi automata on infinite words are essentially Büchi automata on $\{1\}$-trees.

Again, a key issue is testing emptiness.

PROPOSITION 5. [62, 82] *The nonemptiness problem for nondeterministic Büchi tree automata is decidable in quadratic time.*

2.5 Alternating Automata on Infinite Trees

An *alternating Büchi tree automaton* A is a tuple $(\Sigma, \mathcal{D}, S, s^0, \rho, F)$. Here Σ is a finite alphabet, $\mathcal{D} \subset N$ is a finite set of arities, S is a finite set of states, $s^0 \in S$ is an initial state, $F \subseteq S$ is a set of accepting states, and $\rho : S \times \Sigma \times \mathcal{D} \to \mathcal{B}^+(N \times S)$ is a partial transition function, where $\rho(s, a, k) \in \mathcal{B}^+(\{1, \ldots, k\} \times S)$ for each $s \in S$, $a \in \Sigma$, and $k \in \mathcal{D}$ such that $\rho(s, a, k)$ is defined. For example, $\rho(s, a, 2) = ((1, s_1) \vee (2, s_2)) \wedge ((1, s_3) \vee (2, s_1))$ means that the automaton can choose between four splitting possibilities. In the first possibility, one copy proceeds in direction 1 in the state s_1 and one copy proceeds in direction 1 in the state s_3. In the second possibility, one copy proceeds in direction 1 in the state s_1 and one copy proceeds in direction 2 in the state s_1. In the third possibility, one copy proceeds in direction 2 in the state s_2 and one copy proceeds in direction 1 in the state s_3. Finally, in the fourth possibility, one copy proceeds in direction 2 in the state s_2 and one copy proceeds in direction 2 in the state s_1. Note that it is possible for more than one copy to proceed in the same direction.

A run r of an alternating Büchi tree automaton A on a Σ-labeled leafless $\mathcal{D}$-tree $\langle \tau, \mathcal{T}\rangle$ is a $\tau \times S$-labeled tree. Each node of r corresponds to a node of τ. A node in r, labeled by (x, s), describes a copy of the automaton that reads the node x of τ in the state s.

Note that many nodes of r can correspond to the same node of τ; in contrast, in a run of a nondeterministic automaton on $\langle \tau, \mathcal{T} \rangle$ there is a one-to-one correspondence between the nodes of the run and the nodes of the tree. The labels of a node and its children have to satisfy the transition function. Formally, r is a Σ_r-labeled tree $\langle \tau_r, \mathcal{T}_r \rangle$ where $\Sigma_r = \tau \times S$ and $\langle \tau_r, \mathcal{T}_r \rangle$ satisfies the following:

1. $\mathcal{T}_r(\varepsilon) = (\varepsilon, s^0)$.

2. Let $y \in \tau_r$, $\mathcal{T}_r(y) = (x, s)$, $arity(x) = k$, and $\rho(s, \mathcal{T}(x), k) = \theta$. Then there is a set $Q = \{(c_1, s_1), (c_1, s_1), \ldots, (c_n, s_n)\} \subseteq \{1, \ldots, k\} \times S$ such that

   - Q satisfies θ, and
   - for all $1 \leq i \leq n$, we have $y \cdot i \in \tau_r$ and $\mathcal{T}_r(y \cdot i) = (x \cdot c_i, s_i)$.

For example, if $\langle \tau, \mathcal{T} \rangle$ is a tree with $arity(\varepsilon) = 2$, $\mathcal{T}(\varepsilon) = a$ and $\rho(s^0, a) = ((1, s_1) \vee (1, s_2)) \wedge ((1, s_3) \vee (1, s_1))$, then the nodes of $\langle \tau_r, \mathcal{T}_r \rangle$ at level 1 include the label $(1, s_1)$ or $(1, s_2)$, and include the label $(1, s_3)$ or $(1, s_1)$.

As with alternating Büchi automata on words, alternating Büchi tree automata are as expressive as nondeterministic Büchi tree automata.

PROPOSITION 6. [52, 56] *Let A be an alternating Büchi tree automaton with n states. Then there is a nondeterministic Büchi tree automaton A_n with 3^n states such that $T_\omega(A_n) = T_\omega(A)$.*

By combining Propositions 5 and 6 (with its exponential blowup), we can obtain a nonemptiness test for alternating Büchi tree automata.

PROPOSITION 7. *The nonemptiness problem for alternating Büchi tree automata is decidable in exponential time.*

Does the size of the alphabet affect the complexity of the nonemptiness problem? For nondeterministic tree automata, the nonemptiness problem is reducible to the 1-letter nonemptiness problem, that is, to the nonemptiness problem for nondeterministic tree automata over 1-letter alphabets (i.e., $|\Sigma| = 1$). Indeed, instead checking the nonemptiness of an automaton $A = (\Sigma, \mathcal{D}, S, S^0, \rho, F)$, one can check the nonemptiness of the automaton $A' = (\{a\}, \mathcal{D}, S, S^0, \rho', F)$ where for all $s \in S$, we have $\rho'(s, a, k) = \bigcup_{a \in \Sigma} \rho(s, a, k)$. It is easy to see that A accepts some tree iff A' accepts some a-labeled tree. This can be viewed as if A' first guesses a Σ-labeling for the input tree and then proceeds like A on this Σ-labeled tree.

This reduction is not valid for alternating tree automata. Suppose that we defined A' by taking $\rho'(s, a, k) = \bigvee_{a \in \Sigma} \rho(s, a, k)$. Then, if A' accepts some a-labeled tree, it still does not guarantee that A accepts some tree. A necessary condition for the validity of the reduction is that different copies of A' that run on the same subtree guess the same Σ-labeling for this subtree. Nothing, however, prevents one copy of A' to proceed according to one labeling and another copy to proceed according to a different labeling. This problem does not occur when A is defined over a singleton alphabet. There, it is guaranteed that all copies proceed according to the same (single) labeling.

As we see later, in our applications we sometimes do have 1-letter alphabets, which makes the 1-letter nonemptiness problem for alternating automata of interest. It turns out that this problem is easier than the general nonemptiness problem. Actually, it is

as easy as the nonemptiness problem for nondeterministic Büchi tree automata (Proposition 5). This result requires also *uniformity*, i.e., $|\mathcal{D}| = 1$.

PROPOSITION 8. [42] *The 1-letter nonemptiness problem for uniform alternating Büchi tree automata is decidable in quadratic time.*

As we shall see later, the alternating automata in our applications have a special structure, studied first in [54]. A *weak alternating tree automaton* (WAA) is an alternating Büchi tree automaton in which there exists a partition of the state set S into disjoint sets $S_1, \ldots, S_n$ such that for each set S_i, either $S_i \subseteq F$, in which case S_i is an *accepting set*, or $S_i \cap F = \emptyset$, in which case S_i is a *rejecting set*. In addition, there exists a partial order $\leq$ on the collection of the S_i's such that for every $s \in S_i$ and $s' \in S_j$ for which s' occurs in $\rho(s, a, k)$, for some $a \in \Sigma$ and $k \in \mathcal{D}$, we have $S_j \leq S_i$. Thus, transitions from a state in S_i lead to states in either the same S_i or a lower one. It follows that every infinite path of a run of a WAA ultimately gets "trapped" within some S_i. The path then satisfies the acceptance condition if and only if S_i is an accepting set. That is, a run visits infinitely many states in F if and only if it gets trapped in an accepting set. The number of sets in the partition of S is defined as the *depth* of the automaton.

It turns out that the nonemptiness problem for WAA on 1-letter alphabets is easier than nonemptiness problem for alternating Büchi automata on 1-letter alphabets.

PROPOSITION 9. [42] *The 1-letter nonemptiness problem for uniform weak alternating tree automata is decidable in linear time.*

As we shall see, the WAA that we use have an even more special structure. In these WAA, each set S_i can be classified as either *transient*, *existential*, or *universal*, such that for each set S_i and for all $s \in Q_i$, $a \in \Sigma$, and $k \in \mathcal{D}$, the following hold:

1. If S_i is transient, then $\rho(s, a, k)$ contains no elements of S_i.
2. If S_i is existential, then $\rho(s, a, k)$ only contains *disjunctively related* elements of S_i (i.e. if the transition is rewritten in disjunctive normal form, there is at most one element of S_i in each disjunct).
3. If Q_i is universal, then $\rho(s, a, k)$ only contains *conjunctively related* elements of S_i (i.e. if the transition is rewritten in conjunctive normal form, there is at most one element of Q_i in each conjunct).

This means that it is only when moving from one S_i to the next, that alternation actually occurs (alternation is moving from a state that is conjunctively related to states in its set to a state that is disjunctively related to states in its set, or vice-versa). In other words, when a copy of the automaton visits a state in some existential set S_i, then as long as it stays in this set, it proceeds in an "existential mode"; namely, it imposes only existential requirement on its successors in S_i. Similarly, when a copy of the automaton visits a state in some universal set S_i, then as long as it stays in this set, it proceeds in a "universal mode". Thus, whenever a copy alternates modes, it must be that it moves from one S_i to the next. We call a WAA that satisfies this property a *limited-alternation*[1] WAA.

PROPOSITION 10. [42] *The 1-letter nonemptiness problem for uniform limited-alternation WAA of size n and depth m can be solved in space $O(m \log^2 n)$.*

[1]The term used in [42] is *hesitant*.

3 TEMPORAL LOGICS AND ALTERNATING AUTOMATA

3.1 Linear Temporal Logic

Formulas of *linear temporal logic* (LTL) are built from a set *Prop* of atomic propositions and are closed under the application of Boolean connectives, the unary temporal connective X (*next*), and the binary temporal connectives U (*until*) and R (*release*) [22]. LTL is interpreted over *computations.* A computation is a function $\pi : \omega \to 2^{Prop}$, which assigns truth values to the elements of *Prop* at each time instant (natural number). (This corresponds to using the ordered natural numbers as the frame; see Chapter 11). A computation π and a point $i \in \omega$ satisfies an LTL formula φ, denoted $\pi, i \models \varphi$, under the following conditions:

- $\pi, i \models p$ for $p \in Prop$ iff $p \in \pi(i)$.
- $\pi, i \models \xi \wedge \psi$ iff $\pi, i \models \xi$ and $\pi, i \models \psi$.
- $\pi, i \models \neg\varphi$ iff not $\pi, i \models \varphi$
- $\pi, i \models X\varphi$ iff $\pi, i+1 \models \varphi$.
- $\pi, i \models \xi U \psi$ iff for some $j \geq i$, we have $\pi, j \models \psi$ and for all k, $i \leq k < j$, we have $\pi, k \models \xi$.[2]
- $\pi, i \models \xi R \psi$ iff for all $j \geq i$, if $\pi, j \not\models \psi$, then for some k, $i \leq k < j$, we have $\pi, k \models \xi$.

Note that $\neg(X\varphi)$ is equivalent to $X(\neg\varphi)$ and $\neg(\xi U \psi)$ is equivalent to $(\neg\xi) R \psi$. This implies that we can assume that formulas are in *positive normal form*, in which negations are applied only to atomic propositions. This normal form is obtained by pushing negations inward as far as possible, using De Morgan's laws and dualities as above. For example, the formula $G(\neg\mathsf{request} \vee (\mathsf{request} U \mathsf{grant}))$ says that whenever a request is made it holds continuously until it is eventually granted. We say that π *satisfies* a formula φ, denoted $\pi \models \varphi$, iff $\pi, 0 \models \varphi$.

Computations can also be viewed as infinite words over the alphabet 2^{Prop}. It turns out that the computations satisfying a given formula are exactly those accepted by some finite automaton on infinite words. The following theorem establishes a very simple translation between LTL and alternating Büchi automata on infinite words.

THEOREM 11. [53, 74] *Given an LTL formula φ, one can build an alternating Büchi automaton $A_\varphi = (\Sigma, S, s^0, \rho, F)$, where $\Sigma = 2^{Prop}$ and $|S|$ is in $O(|\varphi|)$, such that $L_\omega(A_\varphi)$ is exactly the set of computations satisfying the formula φ.*

Proof. The set S of states consists of all subformulas of φ. The initial state s^0 is φ itself. The set F of accepting states consists of all formulas in S of the form $(\xi R \psi)$. It remains to define the transition function ρ.

- $\rho(p, a) = \mathbf{true}$ if $p \in a$,
- $\rho(\neg p, a) = \mathbf{true}$ if $p \notin a$,

[2]Note that our U operator is not strict; cf. Chapter 11.

- $\rho(p, a) = \mathbf{false}$ if $p \notin a$,
- $\rho(\neg p, a) = \mathbf{false}$ if $p \in a$,
- $\rho(\xi \wedge \psi, a) = \rho(\xi, a) \wedge \rho(\psi, a)$,
- $\rho(\xi \vee \psi, a) = \rho(\xi, a) \vee \rho(\psi, a)$,
- $\rho(X\psi, a) = \psi$,
- $\rho(\xi U \psi, a) = \rho(\psi, a) \vee (\rho(\xi, a) \wedge \xi U \psi)$.
- $\rho(\xi R \psi, a) = \rho(\psi, a) \wedge (\rho(\xi, a) \vee \xi R \psi)$.

❑

By applying Proposition 3, we now get:

COROLLARY 12. [83] *Given an LTL formula φ, one can build a nondeterministic Büchi automaton $A_\varphi = (\Sigma, S, s^0, \rho, F)$, where $\Sigma = 2^{Prop}$ and $|S|$ is in $2^{O(|\varphi|)}$, such that $L_\omega(A_\varphi)$ is exactly the set of computations satisfying the formula φ.*

For a description of optimized translations from LTL to automata, seefor example, [32]. While our focus here is on LTL, the automata-theoretic approach applies also to more expressive, recent industrial specification languages such as ForSpec [2]; see [1, 10].

3.2 Branching Temporal Logic

The branching temporal logic CTL (Computation Tree Logic) provides temporal connectives that are composed of a path quantifier immediately followed by a single linear temporal connective [22]. The path quantifiers are A ("for all paths") and E ("for some path"). The linear-time connectives are X, U, and R. Thus, given a set $Prop$ of atomic propositions, a CTL formula in positive normal form is one of the following:

- p or $\neg p$, for all $p \in AP$,
- $\xi \wedge \psi$ or $\xi \vee \psi$, where ξ and ψ are CTL formulas.
- $EX\xi$, $AX\xi$, $E(\xi U \psi)$, $A(\xi U \psi)$, $E(\xi R \psi)$, or $A(\xi R \psi)$, where ξ and ψ are CTL formulas.

The semantics of CTL is defined with respect to *programs*. A program over a set $Prop$ of atomic propositions is a structure of the form $P = (W, w^0, R, V)$, where W is a set of states, $w^0 \in W$ is an initial state, $R \subseteq W^2$ is a total accessibility relation (i.e., every state can access at least one state), and $V : W \rightarrow 2^{Prop}$ assigns truth values to propositions in $Prop$ for each state in W. The intuition is that W describes all the states that the program could be in (where a state includes the content of the memory, registers, buffers, program counter, etc.), R describes all the possible transitions between states (allowing for nondeterminism), and V relates the states to the propositions (e.g., it tells us in what states the proposition `request` is true); see [47] for a discussion on modelling programs. The assumption that R is total (i.e., that every state has an R-successor) is for technical convenience. We can view a terminated execution as repeating forever its last state. We

say that P is a *finite-state* program if W is finite. A *path* in P is a sequence of states, $\mathbf{u} = u_0, u_1, \ldots$ such that for every $i \geq 0$, we have that $u_i R u_{i+1}$ holds. Such a path is called a u_0-path.

A program $P = (W, w^0, R, V)$ and a state $u \in W$ satisfies a CTL formula φ, denoted $P, u \models \varphi$, under the following conditions:

- $P, u \models p$ for $p \in Prop$ if $p \in V(u)$.
- $P, u \models \neg p$ for $p \in Prop$ if $p \notin V(u)$.
- $P, u \models \xi \wedge \psi$ iff $P, u \models \xi$ and $P, u \models \psi$.
- $P, u \models \xi \vee \psi$ iff $P, u \models \xi$ or $P, u \models \psi$.
- $P, u \models EX\varphi$ if $P, v \models \varphi$ for some v such that uRv holds.
- $P, u \models AX\varphi$ if $P, v \models \varphi$ for all v such that uRv holds.
- $P, u \models E(\xi U \psi)$ if there exist a u-path π such that $\pi, 0 \models \xi U \psi$.
- $P, u \models A(\xi U \psi)$ if for every u-path π we have $\pi, 0 \models \xi U \psi$.
- $P, u \models E(\xi R \psi)$ if there exist a u-path π such that $\pi, 0 \models \xi R \psi$.
- $P, u \models A(\xi R \psi)$ if for every u-path π we have $\pi, 0 \models \xi R \psi$.

For example, the formula $AG(\mathsf{request} \rightarrow EF\mathsf{grant})$ says that whenever a request is made it is eventually granted in some possible future. We say that p satisfies φ, denoted $P \models \varphi$, if $P, w^0 \models \varphi$.

A program $P = (W, w^0, R, V)$ is a *tree program* if (W, R) is a tree and w^0 is its root. Note that in this case P is a leafless 2^{Prop}-labeled tree (it is leafless, since R is total). P is a $\mathcal{D}$-tree program, for $\mathcal{D} \subset N$, if (W, R) is a $\mathcal{D}$-tree. It turns out that the tree programs satisfying a given formula are exactly those accepted by some finite tree automaton. The following theorem establishes a very simple translation between CTL and weak alternating Büchi tree automata.

THEOREM 13. [42, 53] *Given a CTL formula φ and a finite set $\mathcal{D} \subset N$, one can build a limited-alternation WAA $A_\varphi = (\Sigma, \mathcal{D}, S, s^0, \rho, F)$, where $\Sigma = 2^{Prop}$ and $|S|$ is in $O(|\varphi|)$, such that $T_\omega(A_\varphi)$ is exactly the set of $\mathcal{D}$-tree programs satisfying φ.*

Proof. The set S of states consists of all subformulas of φ. The initial state s^0 is φ itself. The set F of accepting states consists of all formulas in S of the form $E(\xi R \psi)$ and $A(\xi R \psi)$. It remains to define the transition function ρ. In the following definition we use the notion of dual, defined in the proof of Theorem 11.

- $\rho(p, a, k) = \mathbf{true}$ if $p \in a$.
- $\rho(\neg p, a, k) = \mathbf{true}$ if $p \notin a$.
- $\rho(p, a, k) = \mathbf{false}$ if $p \notin a$.
- $\rho(\neg p, a, k) = \mathbf{false}$ if $p \in a$.
- $\rho(\xi \wedge \psi, a, k) = \rho(\xi, a, k) \wedge \rho(\psi, a, k)$.

- $\rho(\xi \vee \psi, a, k) = \rho(\xi, a, k) \vee \rho(\psi, a, k)$.
- $\rho(EX\psi, a, k) = \bigvee_{c=0}^{k-1}(c, \psi)$.
- $\rho(AX\psi, a, k) = \bigwedge_{c=0}^{k-1}(c, \psi)$.
- $\rho(E(\xi U \psi), a, k) = \rho(\psi, a, k) \vee (\rho(\xi, a, k) \wedge \bigvee_{c=0}^{k-1}(c, E(\xi U \psi)))$.
- $\rho(A(\xi U \psi), a, k) = \rho(\psi, a, k) \vee (\rho(\xi, a, k) \wedge \bigwedge_{c=0}^{k-1}(c, A(\xi U \psi)))$.
- $\rho(E(\xi R \psi), a, k) = \rho(\psi, a, k) \wedge (\rho(\xi, a, k) \vee \bigvee_{c=0}^{k-1}(c, E(\xi R \psi)))$.
- $\rho(A(\xi U \psi), a, k) = \rho(\psi, a, k) \wedge (\rho(\xi, a, k) \vee \bigwedge_{c=0}^{k-1}(c, A(\xi U \psi)))$.

Finally, we define a partition of S into disjoint sets and a partial order over the sets. Each formula $\psi \in S$ constitutes a (singleton) set $\{\psi\}$ in the partition. The partial order is then defined by $\{\xi\} \leq \{\psi\}$ iff ξ a subformula of ψ. Here, all sets are transient, expect for sets of the form $\{E(\xi U \psi)\}$ and $\{\neg A(\xi U \psi)\}$, which are existential, and sets of the form $\{A(\xi U \psi)\}$ and $\{\neg E(\xi U \psi)\}$, which are universal. Thus, A_φ is a limited-alternation WAA. ❑

While temporal logic was introduced in both a branching-time setting and linear-time setting [59], its introduction to computer science was in a linear-time setting [57], to be followed soon by a branching-time setting [14]. The debate in the computer-science literature regarding the relative merits of the linear-time and branching-time goes back to the early 1980s [3, 13, 23, 25, 28, 43, 58, 76, 78]. For a description of earlier discussions of this distinction in the philosophical-logic literature, see [9]. For a more recent account, see [79].

4 MODEL CHECKING

In this section we focus on model checking finite-state programs. Our computational-complexity results are stated in terms of the size of the programs and the temporal properties being checked. The size $|\varphi|$ of a temporal formula φ is simply its length (as a character string). The size of a finite-state program $P = (W, w^0, R, V)$ is the size of its encoding, which is proportional to the number of states in W and the number of transitions in R. Of course, the size of P can be rather large. For example, if P is a computer circuit with n memory bits, then there are 2^n possible states. In general, the number of states of a program is at least exponential in the size of its description by means of a programing language or a hardware-description languages. This blow-up is referred to as the *state-explosion problem*. Much of the research on model checking is focused on dealing with the state-explosion problem, including on-the-fly search techniques, which search through the state space in a demand-driven fashion [36], and symbolic techniques, which represent large state spaces compactly [4]. See [16, 37] for extensive discussions.

4.1 Linear Temporal Logic

We assume that we are given a finite-state program and an LTL formula that specifies the legal computations of the program. The problem is to check whether all computations of the program are legal.

Let $\mathbf{u} = w_0, w_1 \ldots$ be a w_0-path of a finite-state program $P = (W, w^0, R, V)$. The sequence $V(w_0), V(w_1) \ldots$ is a *computation of* P (note that such a sequence can indeed be viewed as a function $\pi_{\mathbf{u}} : \omega \to 2^{Prop}$, which is how we described computations earlier). We say that P *satisfies* an LTL formula φ if *all* computations of P satisfy φ. The *LTL verification problem* is to check whether P satisfies φ.

We now describe the automata-theoretic approach to the LTL verification problem. A finite-state program $P = (W, w^0, R, V)$ can be viewed as a nondeterministic Büchi automaton $A_P = (\Sigma, W, \{w^0\}, \rho, W)$, where $\Sigma = 2^{Prop}$ and $v \in \rho(u, a)$ iff uRv holds and $a = V(u)$. As this automaton has a set of accepting states equal to the whole set of states, any infinite run of the automaton is accepting. Thus, $L_\omega(A_P)$ is the set of computations of P.

Hence, for a finite-state program P and an LTL formula φ, the verification problem is to verify that all infinite words accepted by the automaton A_P satisfy the formula φ. By Corollary 12, we know that we can build a nondeterministic Büchi automaton A_φ that accepts exactly the computations satisfying the formula φ. The verification problem thus reduces to the automata-theoretic problem of checking that all computations accepted by the automaton A_P are also accepted by the automaton A_φ, that is $L_\omega(A_P) \subseteq L_\omega(A_\varphi)$. Equivalently, we need to check that the automaton that accepts $L_\omega(A_P) \cap L_\omega(\overline{A_\varphi})$ is empty, where $L_\omega(\overline{A_\varphi}) = \overline{L_\omega(A_\varphi)} = \Sigma^\omega - L_\omega(A_\varphi)$.

First, note that, by Corollary 12, $L_\omega(\overline{A_\varphi}) = L_\omega(A_{\neg\varphi})$ and the automaton $A_{\neg\varphi}$ has $2^{O(|\varphi|)}$ states. (A straightforward approach, starting with the automaton A_φ and then complementing it, would result in a doubly exponential blow-up, since complementation of nondeterministic Büchi automata is exponential [41, 50]). To get the intersection of the two automata, we use Proposition 1. Consequently, we can build an automaton for $L_\omega(A_P) \cap L_\omega(A_{\neg\varphi})$ having $|W| \cdot 2^{O(|\varphi|)}$ states. We need to check this automaton for emptiness. Using Proposition 2, we get the following results.

THEOREM 14. [44, 67, 81] *Checking whether a finite-state program P satisfies an LTL formula φ can be done in time $O(|P| \cdot 2^{O(|\varphi|)})$ or in space $O((|\varphi| + \log |P|)^2)$.*

We note that a time upper bound that is polynomial in the size of the program and exponential in the size of the specification is considered here to be reasonable, since the specification is usually rather short [44]. For practical verification algorithms that are based on the automata-theoretic approach see [8, 18].

4.2 Branching Temporal Logic

For linear temporal logic, each program may correspond to infinitely many computations. Model checking is thus reduced to checking inclusion between the set of computations allowed by the program and the language of an automaton describing the formula. For branching temporal logic, each program corresponds to a single “computation tree”. On that account, model checking is reduced to checking acceptance of this computation tree by the automaton describing the formula.

A program $P = (W, w^0, R, V)$ can be viewed as a W-labeled tree $\langle \tau_P, \mathcal{T}_P \rangle$ that corresponds to the unwinding of P from w^0. For every node $w \in W$, let $arity(w)$ denote the number of R-successors of w and let $succ_R(w) = \langle w_1, \ldots, w_{arity(w)} \rangle$ be an ordered list of w's R-successors (we assume that the nodes of W are ordered). τ_P and $\mathcal{T}_P$ are defined inductively:

1. $\varepsilon \in \tau_P$ and $\mathcal{T}_P(\varepsilon) = w^0$.

2. For $y \in \tau_P$ with $succ_R(\mathcal{T}_P(y)) = \langle w_1, \ldots, w_k \rangle$ and for all $1 \leq i \leq k$, we have $y \cdot i \in \tau_P$ and $\mathcal{T}_P(y \cdot i) = w_i$.

Let $\mathcal{D}$ be the set of arities of states of P, i.e., $\mathcal{D} = \{arity(w) \ : \ w \in W\}$. Clearly, τ_P is a $\mathcal{D}$-tree. If P is finite, then $\mathcal{D}$ is finite.

Let $\langle \tau_P, V \cdot \mathcal{T}_P \rangle$ be the 2^{Prop}-labeled $\mathcal{D}$-tree defined by $V \cdot \mathcal{T}_P(y) = V(\mathcal{T}_P(y))$ for $y \in \tau_P$. Let φ be a CTL formula. Suppose that $A_{\mathcal{D},\varphi}$ is an alternating automaton that accepts exactly all $\mathcal{D}$-tree programs that satisfy φ. It can easily be shown that $\langle \tau_P, V \cdot \mathcal{T}_P \rangle$ is accepted by $A_{\mathcal{D},\varphi}$ iff $P \models \varphi$. We now show that by taking the product of P and $A_{\mathcal{D},\varphi}$ we get an alternating Büchi tree automaton on a 1-letter alphabet that is empty iff $\langle \tau_P, V \cdot \mathcal{T}_P \rangle$ is accepted by $A_{\mathcal{D},\varphi}$.

Let $A_{\mathcal{D},\varphi} = (2^{AP}, \mathcal{D}, S, \varphi, \rho, F)$ be a limited-alternation WAA that accepts exactly all $\mathcal{D}$-tree programs that satisfy φ, and let $S_1, \ldots, S_n$ be the partition of S. The *product automaton* of P and $A_{\mathcal{D},\varphi}$ is the limited-alternation WAA $A_{P,\varphi} = (\{a\}, \mathcal{D}, W \times S, \delta, \langle w^0, \varphi \rangle, G)$, where δ and G are defined as follows:

- Let $s \in S$, $w \in W$, $succ_R(w) = \langle w_1, \ldots, w_k \rangle$, and $\rho(s, V(w), k) = \theta$. Then $\delta(\langle w, s \rangle, a, k) = \theta'$, where θ' is obtained from θ by replacing each atom (c, s') in θ by the atom $(c, \langle w_c, s' \rangle)$.

- $G = W \times F$

- $W \times S$ is partitioned to $W \times S_1, W \times S_2, \ldots, W \times S_n$.

- $W \times S_i$ is transient (resp., existential, universal) if S_i is transient (resp., existential, universal), for $1 \leq i \leq n$.

Note that if P has m_1 states and $A_{\mathcal{D},\varphi}$ has m_2 states then $A_{P,\varphi}$ has $O(m_1 m_2)$ states.

PROPOSITION 15. *$A_{P,\varphi}$ is nonempty if and only if $P \models \varphi$.*

We can now put together Propositions 9, 10, and 15 to get a model-checking algorithm for CTL.

THEOREM 16. [15, 42] *Checking whether a finite-state program P satisfies a CTL formula φ can be done in time $O(|P| \cdot |\varphi|)$ or in space $O(|\varphi| \log^2 |P|)$.*

See also [8, 84] for description of practical algorithms. For an extension of automata-theoretic branching-time model checking to more expressive branching-time logics, such as the branching-time logic CTL*, which merges CTL and LTL and is more expressive than both [25], or the modal fixpoint logic, which is more expressive than CTL* [39], see [42].

5 VALIDITY CHECKING

5.1 Linear Temporal Logic

We are given an LTL formula φ. We say that φ is *valid* iff it is true in all computations. By Corollary 12, we know that we can build a nondeterministic Büchi automaton A_φ that accepts exactly the computations in which φ is true. In other words, φ is valid iff $L_\omega(A_\varphi) = \Sigma^\omega$, where $\Sigma = 2^{Prop}$, which holds iff $\Sigma^\omega - L_\omega(A_\varphi) = \emptyset$. Since $\Sigma^\omega - L_\omega(A_\varphi) = L_\omega(A_{\neg\varphi})$, we have that φ is valid iff $L_\omega(A_{\neg\varphi}) = \emptyset$. Thus, validity checking is been reduced to emptiness checking. We can now combine Proposition 2 with Corollary 12:

THEOREM 17. [67] *Checking whether an LTL formula φ is valid can be done in time $O(2^{O(|\varphi|)})$ or in space $O((|\varphi|)^2)$.*

We note that the upper space bound of Theorem 17 is essentially optimal, since the validity problem for LTL is PSPACE-hard [67].

5.2 Branching Temporal Logic

We are given a CTL formula φ. we say that φ is *valid* iff it is true in all programs. For LTL, Theorems 11 and 12 provided automata-theoretic characterizations of all models of the formula. This is not the case for CTL, as Theorem 13 provides only a characterization of tree models. Fortunately, this suffices for validity checking due to the following proposition.

PROPOSITION 18. [22] *Let φ be a CTL formula. Then φ is valid iff φ is true in all $|\varphi|$-tree programs.*

Let A_φ be the automaton $A_\varphi^{\{|\varphi|\}}$, i.e., it is the automaton $A_\varphi^{\mathcal{D}}$ of Theorem 13, with $\mathcal{D} = \{|\varphi|\}$. It follows from Proposition 18 that a CTL formula is valid iff $T_\omega(A_{\neg\varphi}) = \emptyset$. Combining this with Proposition 7, we get:

THEOREM 19. [24] *Checking whether a CTL formula φ is valid can be done in time $O(2^{O(|\varphi|)})$.*

We note that the upper time bound of Theorem 19 is essentially optimal, since the validity problem for CTL is EXPTIME-hard [31]. For a practical algorithm to decide CTL validity, see [49]. For extension of automata-theoretic validity checking to expressive modal logics, see [40, 64, 77].

ACKNOWLEDGEMENTS

Work supported in part by NSF grants CCR-9988322, CCR-0124077, CCR-0311326, IIS-9908435, IIS-9978135, EIA-0086264, and ANI-0216467, by BSF grant 9800096, by Texas ATP grant 003604-0058-2003, and by a grant from the Intel Corporation. Part of this work was done while the author was visiting the Isaac Newton Institute for Mathematical Science, Cambridge, UK, as part of a Special Programme on Logic and Algorithms.

BIBLIOGRAPHY

[1] R. Armoni, D. Bustan, O. Kupferman, and M.Y. Vardi. Resets vs. aborts in linear temporal logic. In *Proc. 9th International Conference on Tools and algorithms for the construction and analysis of systems*, number 2619 in Lecture Notes in Computer Science, pages 65 – 80. Springer-Verlag, 2003.

[2] R. Armoni, L. Fix, A. Flaisher, R. Gerth, B. Ginsburg, T. Kanza, A. Landver, S. Mador-Haim, E. Singerman, A. Tiemeyer, M.Y. Vardi, and Y. Zbar. The ForSpec temporal logic: A new temporal property-specification logic. In *Proc. 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 2280 of *Lecture Notes in Computer Science*, pages 296–211, Grenoble, France, April 2002. Springer-Verlag.

[3] M. Ben-Ari, A. Pnueli, and Z. Manna. The temporal logic of branching time. *Acta Informatica*, 20:207–226, 1983.

[4] R.E. Bryant. Graph-based algorithms for boolean-function manipulation. *IEEE Trans. on Computers*, C-35(8), 1986.

[5] J.A. Brzozowski and E. Leiss. Finite automata and sequential networks. *Theoretical Computer Science*, 10:19–35, 1980.

[6] J.R. Büchi. Weak second-order arithmetic and finite automata. *Zeit. Math. Logik und Grundl. Math.*, 6:66–92, 1960.

[7] J.R. Büchi. On a decision method in restricted second order arithmetic. In *Proc. International Congress on Logic, Method, and Philosophy of Science. 1960*, pages 1–12, Stanford, 1962. Stanford University Press.

[8] J.R. Burch, E.M. Clarke, K.L. McMillan, D.L. Dill, and L.J. Hwang. Symbolic model checking: 10^{20} states and beyond. *Information and Computation*, 98(2):142–170, June 1992.

[9] J. P. Burgess. Logic and time. *J. Symbolic Logic*, 44:566–582, 1979.

[10] D. Bustan, A. Flaisher, O. Grumberg, O. Kupferman, and M.Y. Vardi. Regular vacuity. In *Proc. 13th Advanced Research Working Conference on Correct Hardware Design and Verification Methods*, volume 3725 of *Lecture Notes in Computer Science*, pages 191–206. Springer-Verlag, 2005.

[11] A.K. Chandra, D.C. Kozen, and L.J. Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133, January 1981.

[12] Y. Choueka. Theories of automata on ω-tapes: A simplified approach. *Journal of Computer and System Sciences*, 8:117–141, 1974.

[13] E.M. Clarke and I.A. Draghicescu. Expressibility results for linear-time and branching-time logics. In J.W. de Bakker, W.P. de Roever, and G. Rozenberg, editors, *Proc. Workshop on Linear Time, Branching Time, and Partial Order in Logics and Models for Concurrency*, volume 354 of *Lecture Notes in Computer Science*, pages 428–437. Springer-Verlag, 1988.

[14] E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Proc. Workshop on Logic of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer-Verlag, 1981.

[15] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, January 1986.

[16] E.M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

[17] R. Cleaveland. A linear-time model-checking algorithm for the alternation-free modal μ-calculus. *Formal Methods in System Design*, 2:121–147, 1993.

[18] C. Courcoubetis, M.Y. Vardi, P. Wolper, and M. Yannakakis. Memory efficient algorithms for the verification of temporal properties. *Formal Methods in System Design*, 1:275–288, 1992.

[19] J. E. Doner. Decidability of the weak second-order theory of two successors. *Notices Amer. Math. Soc.*, 12:819, 1965.

[20] C. Elgot. Decision problems of finite-automata design and related arithmetics. *Trans. Amer. Math. Soc.*, 98:21–51, 1961.

[21] E.A. Emerson. Automata, tableaux, and temporal logics. In *Proc. Workshop on Logic of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 79–87. Springer-Verlag, 1985.

[22] E.A. Emerson. Temporal and modal logic. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 16, pages 997–1072. Elsevier, MIT Press, 1990.

[23] E.A. Emerson and E.M. Clarke. Characterizing correctness properties of parallel programs using fixpoints. In *Proc. 7th InternationalColloq. on Automata, Languages and Programming*, pages 169–181, 1980.

[24] E.A. Emerson and J.Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, 30:1–24, 1985.

[25] E.A. Emerson and J.Y. Halpern. Sometimes and not never revisited: On branching versus linear time. *Journal of the ACM*, 33(1):151–178, 1986.

[26] E.A. Emerson and C. Jutla. The complexity of tree automata and logics of programs. In *Proc. 29th IEEE Symp. on Foundations of Computer Science*, pages 328–337, White Plains, October 1988.

[27] E.A. Emerson and C. Jutla. Tree automata, μ-calculus and determinacy. In *Proc. 32nd IEEE Symp. on Foundations of Computer Science*, pages 368–377, San Juan, October 1991.

[28] E.A. Emerson and C.-L. Lei. Modalities for model checking: Branching time logic strikes back. In *Proc. 20th ACM Symp. on Principles of Programming Languages*, pages 84–96, New Orleans, January 1985.

[29] E.A. Emerson and C.-L. Lei. Temporal model checking under generalized fairness constraints. In *Proc. 18th Hawaii International Conference on System Sciences*, North Holywood, 1985. Western Periodicals Company.

[30] E.A. Emerson and A. P. Sistla. Deciding branching time logic. In *Proc. 16th ACM Symp. on Theory of Computing*, pages 14–24, Washington, April 1984.

[31] M.J. Fischer and R.E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and Systems Sciences*, 18:194–211, 1979.

[32] C. Fritz. Concepts of automata construction from LTL. In *Proc. 12th Int'l Conf. on Logic for Programming, Artificial Intelligence, and Reasoning*, Lecture Notes in Computer Science 3835, pages 728–742. Springer-Verlag, 2005.

[33] D.M. Gabbay. Applications of trees to intermediate logics i. *J. Symbolic Logic*, 37:135–138, 1972.

[34] R. Gerth, D. Peled, M.Y. Vardi, and P. Wolper. Simple on-the-fly automatic verification of linear temporal logic. In P. Dembiski and M. Sredniawa, editors, *Protocol Specification, Testing, and Verification*, pages 3–18. Chapman & Hall, August 1995.

[35] T. Hafer and W. Thomas. Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. In *Proc. 14th International Coll. on Automata, Languages, and Programming*, volume 267 of *Lecture Notes in Computer Science*, pages 269–279. Springer-Verlag, 1987.

[36] G. Holzmann. An improved protocol reachability analysis technique. *Software Practice and Experience*, 18(2):137–161, February 1988.

[37] G.J. Holzmann. *The SPIN Model Checker: Primer and Reference Manual*. Addison-Wesley, 2003.

[38] J.A.W. Kamp. *Tense Logic and the Theory of Order*. PhD thesis, UCLA, 1968.

[39] D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[40] O. Kupferman, U. Sattler, and M. Y. Vardi. The complexity of the graded μ-calculus. In *Proceedings of the Conference on Automated Deduction*, volume 2392 of *Lecture Notes in Artificial Intelligence*, pages 423–437. Springer Verlag, 2002.

[41] O. Kupferman and M.Y. Vardi. Weak alternating automata are not that weak. *ACM Trans. on Computational Logic*, 2(2):408–429, July 2001.

[42] O. Kupferman, M.Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, March 2000.

[43] L. Lamport. Sometimes is sometimes "not never" - on the temporal logic of programs. In *Proc. 7th ACM Symp. on Principles of Programming Languages*, pages 174–185, January 1980.

[44] O. Lichtenstein and A. Pnueli. Checking that finite state concurrent programs satisfy their linear specification. In *Proc. 12th ACM Symp. on Principles of Programming Languages*, pages 97–107, New Orleans, January 1985.

[45] O. Lichtenstein, A. Pnueli, and L. Zuck. The glory of the past. In *Logics of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 196–218, Brooklyn, June 1985. Springer-Verlag.

[46] M.T. Liu. Protocol engineering. *Advances in Computing*, 29:79–195, 1989.

[47] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer-Verlag, Berlin, January 1992.

[48] Z. Manna and A. Pnueli. Temporal specification and verification of reactive modules. Technical report, Weizmann Institute, 1992.

[49] W. Marrero. Using BDDs to decide ctl. In *Proc. 11th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science 3440, pages 222–236. Springer-verlag, 2005.

[50] M. Michel. Complementation is more difficult with automata on infinite words. CNET, Paris, 1988.

[51] S. Miyano and T. Hayashi. Alternating finite automata on ω-words. *Theoretical Computer Science*, 32:321–330, 1984.

[52] A.W. Mostowski. Regular expressions for infinite trees and a standard form of automata. In *Computation Theory*, volume 208 of *Lecture Notes in Computer Science*, pages 157–168. Springer-Verlag, 1984.

[53] D.E. Muller, A. Saoudi, and P. E. Schupp. Weak alternating automata give a simple explanation of why most temporal and dynamic logics are decidable in exponential time. In *Proceedings 3rd IEEE Symp. on Logic in Computer Science*, pages 422–427, Edinburgh, July 1988.

[54] D.E. Muller, A. Saoudi, and P.E. Schupp. Alternating automata, the weak monadic theory of the tree and its complexity. In *Proc. 13th International Colloquium on Automata, Languages and Programming*, volume 226 of *Lecture Notes in Computer Science*, pages 275 – 283. Springer-Verlag, 1986.

[55] D.E. Muller and P.E. Schupp. Alternating automata on infinite trees. *Theoretical Computer Science*, 54:267–276, 1987.

[56] D.E. Muller and P.E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of theorems of Rabin, McNaughton and Safra. *Theoretical Computer Science*, 141:69–107, 1995.

[57] A. Pnueli. The temporal logic of programs. In *Proc. 18th IEEE Symp. on Foundation of Computer Science*, pages 46–57, 1977.

[58] A. Pnueli. Linear and branching structures in the semantics and logics of reactive systems. In *Proc. 12th International Colloquium on Automata, Languages and Programming*, volume 194, pages 15–32. Lecture Notes in Computer Science, Springer-Verlag, 1985.

[59] A. Prior. *Time and Modality.* Oxford University Press, 1957.

[60] J.P. Queille and J. Sifakis. Specification and verification of concurrent systems in Cesar. In *Proc. 5th International Symp. on Programming*, volume 137 of *Lecture Notes in Computer Science*, pages 337–351. Springer-Verlag, 1981.

[61] M.O. Rabin. Decidability of second order theories and automata on infinite trees. *Transaction of the AMS*, 141:1–35, 1969.

[62] M.O. Rabin. Weakly definable relations and special automata. In *Proc. Symp. Math. Logic and Foundations of Set Theory*, pages 1–23. North Holland, 1970.

[63] H. Rudin. Network protocols and tools to help produce them. *Annual Review of Computer Science*, 2:291–316, 1987.

[64] U. Sattler and M.Y. Vardi. The hybrid μ-calculus. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proc. 1st Int'l Joint Conf. on Automated Reasoning*, Lecture Notes in Computer Science 2083, pages 76–91. Springer-Verlag, 2001.

[65] W.J. Savitch. Relationship between nondeterministic and deterministic tape complexities. *Journal on Computer and System Sciences*, 4:177–192, 1970.

[66] A.P. Sistla. *Theoretical issues in the design of distributed and concurrent systems.* PhD thesis, Harward University, Cambridge, MA, 1983.

[67] A.P. Sistla and E.M. Clarke. The complexity of propositional linear temporal logic. *Journal ACM*, 32:733–749, 1985.

[68] A.P. Sistla, M.Y. Vardi, and P. Wolper. The complementation problem for Büchi automata with applications to temporal logic. *Theoretical Computer Science*, 49:217–237, 1987.

[69] R.S. Street and E.A. Emerson. An elementary decision procedure for the μ-calculus. In *Proc. 11th International Colloquium on Automata, Languages and Programming*, volume 172, pages 465–472. Lecture Notes in Computer Science, Springer-Verlag, July 1984.

[70] R.S. Streett. Propositional dynamic logic of looping and converse. *Information and Control*, 54:121–141, 1982.

[71] R.E. Tarjan. Depth first search and linear graph algorithms. *SIAM Journal of Computing*, 1(2):146–160, 1972.

[72] J.W. Thatcher and J.B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical System Theory*, 2:57–81, 1968.

[73] B.A. Trakhtenbrot. Finite automata and monadic second order logic. *Siberian Math. J*, 3:101–131, 1962. Russian; English translation in: AMS Transl. 59 (1966), 23-55.

[74] M.Y. Vardi. Nontraditional applications of automata theory. In *Proc. International Symp. on Theoretical Aspects of Computer Software*, volume 789 of *Lecture Notes in Computer Science*, pages 575–597. Springer-Verlag, 1994.

[75] M.Y. Vardi. Alternating automata and program verification. In *Computer Science Today –Recent Trends and Developments*, volume 1000 of *Lecture Notes in Computer Science*, pages 471–485. Springer-Verlag, Berlin, 1995.

[76] M.Y. Vardi. Linear vs. branching time: A complexity-theoretic perspective. In *Proc. 13th IEEE Sym.. on Logic in Computer Science*, pages 394–405, 1998.

[77] M.Y. Vardi. Reasoning about the past with two-way automata. In *Proc. 25th International Coll. on Automata, Languages, and Programming*, volume 1443 of *Lecture Notes in Computer Science*, pages 628–641. Springer-Verlag, Berlin, July 1998.

[78] M.Y. Vardi. Sometimes and not never re-revisited: on branching vs. linear time. In D. Sangiorgi and R. de Simone, editors, *Proc. 9th Int'l Conf. on Concurrency Theory*, Lecture Notes in Computer Science 1466, pages 1–17, 1998.

[79] M.Y. Vardi. Branching vs. linear time: Final showdown. In *Proc. 7th International Conference on Tools and algorithms for the construction and analysis of systems*, volume 2031 of *Lecture Notes in Computer Science*, pages 1–22. Springer-Verlag, 2001.

[80] M.Y. Vardi and P. Wolper. Yet another process logic. In *Logics of Programs*, volume 164 of *Lecture Notes in Computer Science*, pages 501–512. Springer-Verlag, 1984.

[81] M.Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *Proc. 1st Symp. on Logic in Computer Science*, pages 332–344, Cambridge, June 1986.

[82] M.Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and System Science*, 32(2):182–221, April 1986.
[83] M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, November 1994.
[84] W. Visser and H. Barringer. Practical CTL* model checking: Should SPIN be extended? *International Journal on Software Tools for Technology Transfer*, 2(4):350–365, 2000.
[85] P. Wolper, M.Y. Vardi, and A.P. Sistla. Reasoning about infinite computation paths. In *Proc. 24th IEEE Symp. on Foundations of Computer Science*, pages 185–194, Tucson, 1983.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

18

INTELLIGENT AGENTS AND COMMON SENSE REASONING

John-Jules Meyer and Frank Veltman

1 Introduction . . . 991
2 Intelligent agents . . . 992
3 Epistemic and doxastic logic . . . 993
4 Deontic logic . . . 996
4.1 Standard deontic logic . . . 996
4.2 Dynamic deontic logic . . . 997
5 BDI logic . . . 999
6 KARO logic . . . 1005
7 Multi-agent logics . . . 1010
7.1 Multi-agent epistemic logic . . . 1010
7.2 Multi-agent BDI logic . . . 1013
8 Further developments . . . 1015
9 Counterfactual conditionals . . . 1017
10 Non-monotonic consequence relations . . . 1022
11 Belief revision . . . 1024

1 INTRODUCTION

Modal logic plays an important role in the field of artificial intelligence (AI). This can be understood from the fact that AI tries to capture aspects of human intelligence by formalizing these in such a way that they can be implemented in an artificial system, particularly a computer-based system. This entails the need for formalization of mental attitudes such as beliefs and desires. In philosophical logic one has studied many of these attitudes using modal logic. Therefore it is only natural that AI researchers have resorted to modal logic for the formal description of the mental attitudes of their intended artifacts. (This is not to say that modal logic is the only way to represent these mental attitudes, since some choose to stick to classical predicate logic as closely as possible. Still, it is recognized widely within the AI community that modal logic presents a valuable tool!)

So, in this chapter we will be concerned with the use of modal logic techniques to formally describe mental attitudes of intelligent systems. We have chosen to split our

treatment into two parts. The first part deals with the use of modal logic for the description of so-called *intelligent agents*, an area of AI that emerged at the end of the 80s, dealing with the theory and practice of the construction of autonomous software or hardware entities that act intelligently (rationally). Typical issues are how to deal with motivational attitudes such as intentions and with informational attitudes such as beliefs. Special attention is paid to 'social' attitudes within a 'multi-agent system'. In such an 'agent society' it becomes important to analyze multi-agent informational and motivational attitudes such as common knowledge and collective intention.

The second part is related to this, but the topics discussed are older. An important problem in AI concerns the question as to how to formalize *commonsense reasoning*, the way humans reason 'in daily life', so to speak, as opposed to reasoning in formal sciences such as mathematics and logic. For example, here one is interested in reasoning patterns connected with defaults (rules of thumb) and counterfactuals ('if ... had been the case, then ... would have been the case'). The study of these reasoning mechanisms appeared to be much more difficult than originally anticipated, and has become a major subject of study within AI since the beginning of the 80s. It includes so-called *non-monotonic reasoning* (reasoning in which earlier conclusions can get lost when more premises become available) and *belief revision* (dealing with how the beliefs of a reasoner change when new information becomes available and is incorporated).

Of course, since also in commonsense reasoning notions such as knowledge and belief play an important role, there is a natural relation with the field of intelligent agents, but the emphasis is different. However, as artificial agents become more intelligent and will invade daily life (such as e.g. the application in so-called companion robots that are supposed to assist and entertain elderly people), undoubtedly there will be a moment where these agents should also employ some (possibly restricted) form of commonsense reasoning![1]

2 INTELLIGENT AGENTS

Intelligent agents have become a major field of research in AI. Although there is little consensus about the precise definition of an intelligent agent, it is generally held that agents are *autonomous* pieces of hardware/software, able to take initiative on behalf of a user or, more generally, to satisfy some goal. Agents are often held to possess *mental attitudes*; they are supposed to deal with information, and act upon this, based on motivation. This calls for a description in terms of the agent's beliefs/knowledge, desires, goals, intentions, commitments, obligations, etc. To describe these mental or cognitive attitudes one may fruitfully employ *modal logic*. Typically for the description of agents one needs an amalgam of modal operators/logics to cater for several of the mental attitudes as mentioned above. Moreover, since agents by definition act and display behavior, it is important to include the *dynamics* of these mental attitudes in the description. One might even maintain that the logics of some of these attitudes, such as goal directedness and *a fortiori* desire, have little interest *per se*: they are rather weak logics without exciting properties. What makes them interesting is their dynamics: their change over time in connection with each other! So, although (modal) logics for e.g. knowledge, belief,

[1]The first part (section 2–8) were written by John-Jules Meyer. Frank Veltman is responsible for part 2 (sections 9–11).

desires etc. certainly play a role, it is also imperative to be able to specify the agent's behavior / attitudes over time. Therefore, generally also a (modal) logic of time or action plays a role in agent specification logics. In this section we will first spend some time on modal logics for some of the mental attitudes in isolation, after which we will turn to agent logics proper that are proposed in the literature, which typically are mixtures of these 'single-attitude' logics and contain an element of time and/or action. Our emphasis in this part lies on the presentation of the logical languages and their semantics, and less on axiomatics and metatheory.

3 EPISTEMIC AND DOXASTIC LOGIC

Epistemic logic deals with the mental attitude of knowledge while doxastic logic treats belief. These logics have become quite popular in both computer and artificial intelligence to describe the knowledge/belief involved in (particularly distributed) computation processes and in agents. As to the former, the work of Halpern et al. [31] must be mentioned. In this chapter we concentrate on the role of epistemic/doxastic logic in AI, and on the description of intelligent agents in particular. The modal approach to knowledge/belief is built on the observation that if an agent is not sure about the truth of a certain proposition p (say that it rains outside), it must reckon both with the possibility that p holds and with the possibility that p does not hold. Formally this is captured by a Kripke model in which in the actual world, the agent considers several possible alternatives (captured by the accessibility relation), some of which satisfy p while other ones do not satisfy p. So we have a very intuitive use of modal semantics here: a formula φ is known / believed by the agent if all alternatives deemed possible by the agent (formally, all worlds accessible for the agent from the actual world) satisfy φ.

Thus, we have the following formal definitions. The language is obtained by taking classical (propositional) logic augmented by a clause for the knowledge or belief operator. We assume a set $\mathcal{P}$ of atomic formulas.

DEFINITION 1. (Language of epistemic/doxastic logic)

- Every atomic formula in $\mathcal{P}$ is an epistemic (doxastic) formula;
- if φ_1 and φ_2 are epistemic (doxastic) formulas, then $\neg\varphi_1, \varphi_1 \vee \varphi_2$ are epistemic (doxastic) formulas;
- if φ is an epistemic (doxastic) formula, then $\mathbf{K}\varphi$ ($\mathbf{B}\varphi$) is an epistemic (doxastic) formula.

Other propositional connectives (such as $\wedge, \rightarrow, \leftrightarrow$) are introduced as (the usual) abbreviations.

DEFINITION 2. (Kripke models for epistemic/doxstic languages)
A model for an epistemic/doxastic language is a triple

$$\mathfrak{M} = \langle W, V, R \rangle,$$

where:

- W is a non-empty set of states (or worlds);
- V is a truth assignment function per state;
- R is an accessibility relation on W for interpreting the modal operator $\mathbf{K}$ or $\mathbf{B}$. In the former case R is assumed to be an equivalence relation, while for the latter R is assumed to be euclidean, transitive and serial.

The set of worlds that are accessible from a certain world must be viewed as epistemic alternatives for this world: if the agent is in this world he is not able to distinguish between these accessible worlds due to his (lack of) knowledge/belief; as far he is concerned he could be in any of the alternatives.

The reason that for modelling knowledge the accessibility relation is taken to be an equivalence relation, can be understood as follows: the agent, being in a certain state, considers a set of alternatives which are all alternatives of each other and one of which is the actual state (so the agent considers his true state as an alternative).

For belief this would be too strong: in particular, for belief it is not reasonable to assume that the agent always considers his true state as an alternative, since he may be mistaken. So, for belief, weaker assumptions are assumed, which nevertheless result in a number of interesting validities.

DEFINITION 3. (Interpretation of epistemic / doxastic formulas.)
In order to determine whether an epistemic (doxastic) formula is true in a model/state pair $\mathfrak{M}, w$ (if so, we write $\mathfrak{M}, w \models \varphi$), we stipulate:

- $\mathfrak{M}, w \models p$ iff $V(w)(p) = \mathit{true}$, for $p \in \mathcal{P}$;
- the logical connectives are interpreted as usual;
- $\mathfrak{M}, w \models \mathbf{K}\varphi(\mathbf{B}\varphi)$ iff $\mathfrak{M}, w' \models \varphi$ for all w' with $R(w, w')$.

The last clause can be understood as follows: an agent knows (believes) a formula to be true if the formula is true in all the epistemic alternatives that the agent considers at the state he is in (represented by the accessibility relation).

DEFINITION 4. (Validity)

- A formula φ is verified by a model $\mathfrak{M} = \langle W, V, R \rangle$ iff it is true in all worlds of $\mathfrak{M}$: $\mathfrak{M} \models \varphi \Leftrightarrow \mathfrak{M}, w \models \varphi$ for all $w \in W$.
- A formula is valid iff it is verified by all models of a given form: $\models \varphi \Leftrightarrow \mathfrak{M} \models \varphi$ for all models $\mathfrak{M}$ of the form considered.

Validities in epistemic logic with respect to the given models (which we will refer to the 'axioms' of knowledge) are:

PROPOSITION 5.

- $\models \mathbf{K}(\varphi \to \psi) \to (\mathbf{K}\varphi \to \mathbf{K}\psi)$
- $\models \mathbf{K}\varphi \to \varphi$

- $\models \mathbf{K}\varphi \to \mathbf{KK}\varphi$
- $\models \neg\mathbf{K}\varphi \to \mathbf{K}\neg\mathbf{K}\varphi$

The first validity says that knowledge is closed under implication: if both the implication $\varphi \to \psi$ and the antecedent φ is known then also the conclusion ψ is known. This is of course a very 'idealized' property of knowledge, but its validity is at the very heart of using so-called normal modal logic as we do here.

The second validity expresses that knowledge is true. (One cannot honestly, truthfully and justifiably state to *know* something that is false.) The third and fourth validities express a form of introspection: the agent knows what it knows, in the sense that it knows that it knows something (the second axiom), and, moreover, it knows what it does *not* know (the third axiom). Of course, this may be very unrealistic to assume for some intelligent agents, such as humans, but often it makes sense to assume it in the case of artificial agents, either by virtue of their finitary nature or by way of some idealization. In any case it makes life easier, since the resulting logic, called **S5**, is very elegant (has relatively simple models) and enjoys several pleasant properties ([56]).The logic can be axiomized by taking the four above validities as axioms, together with an axiomatization of classical propositional logic and the rules of Modus Ponens and Necessitation ($\varphi/\mathbf{K}\varphi$).

With respect to doxastic logic we obtain the following validities:

PROPOSITION 6.

- $\models \mathbf{B}(\varphi \to \psi) \to (\mathbf{B}\varphi \to \mathbf{B}\psi)$
- $\models \neg\mathbf{B}\bot$
- $\models \mathbf{B}\varphi \to \mathbf{BB}\varphi$
- $\models \neg\mathbf{B}\varphi \to \mathbf{B}\neg\mathbf{B}\varphi$

Again we observe the introspection properties, but the second validity now states that an agent's belief is not inconsistent, which is weaker than the property that belief should be true. If one takes these properties as axioms completed by modus ponens, necessitation for **B** and (sufficient) classical propositional validities, one obtains the system known as **KD45**.

A natural question is whether the knowledge and belief modalities are interrelated in some meaningful way. The answer is more involved than one might suspect. In the literature (for example [50, 75, 89, 90]), several interesting possibilities for such an interaction have been investigated. In these studies it became clear that one has to be careful in putting several plausible properties together, as one might otherwise end up with undesirable properties such as the collapse of knowledge and belief! As usual with applications of modal logic, one may wonder whether the properties one obtains are all desirable and not 'over-idealizations'. In the realm of epistemic/doxastic logic one may dispute the so-called paradoxes of *logical omniscience*. Most of these are inherent in the use of (normal) modal logic using standard Kripke semantics. For example, the basic modal property $\models \mathbf{K}(\varphi \to \psi) \to (\mathbf{K}\varphi \to \mathbf{K}\psi)$ gives rise to a Sorites-like paradox: the agent knows *all* consequences of its knowledge, and likewise for belief. For 'finitary', resource-bounded agents this is unrealistic. One can now either take this for granted and

view the modal operators for knowledge/belief as idealizations of the real thing, or one has to resort to non-standard ('non-normal') semantics (such as neighborhood semantics) to be able to avoid validities such as the above one. For a fuller treatment of this issue we refer to [56, 54].

4 DEONTIC LOGIC

4.1 Standard deontic logic

One of the first systems for deontic logic that really was a serious attempt to capture deontic reasoning was the now so-called "Old System" of Von Wright ([87]), of which a modal logic (Kripke-style) version has become known as Standard Deontic Logic (**SDL**). The syntax of SDL is that of a propositional modal logic with a modal operator O for obligation. $O\varphi$ is read as 'φ is obligatory / obligated' or 'it ought to be the case that φ'. The modalities F and P for 'it is forbidden' and 'it is permitted', respectively, are introduced as abbreviations: $F\varphi = O\neg\varphi$ and $P\varphi = \neg F\varphi$: something is forbidden iff its negation is obligatory, and something is permitted iff it is not forbidden. **SDL** has a Kripke-style modal semantics based on a set of possible worlds (, a truth assignment function of primitive propositions per possible world) and an accessibility relation associated with the O-modality. This accessibility relation points to "ideal" or "perfect deontic alternatives" of the world under consideration. The crux behind this is that in some possible world something (say φ) is obligated, if φ holds in all the perfect alternatives of this world, as indicated by the accessibility relation.

So, formally these models have the following form: $\mathfrak{M} = (W, V, R_O)$, where W is the set of states/ worlds, V is a truth assignment function, and R_O is the deontic accessibility relation, which is assumed to be serial, i.e. for all $w \in W$ there is a $w' \in W$ such that $R_O(w, w')$.

The operator O is interpreted by means of the relation R_O: $\mathfrak{M}, w \models O\varphi$ iff $M, w' \models \varphi$ for all w' with $R_O(w, w')$. Validity is defined as usual for modal logic. We obtain the following validities:

PROPOSITION 7.

- $O(\varphi \to \psi) \to (O\varphi \to O\psi)$
- $O(\varphi \wedge \psi) \leftrightarrow (O\varphi \wedge O\psi)$
- $P(\varphi \wedge \psi) \to (P\varphi \wedge P\psi)$
- $(F\varphi \vee F\psi) \to F(\varphi \wedge \psi)$
- $(O\varphi \vee O\psi) \to O(\varphi \vee \psi)$
- $P(\varphi \vee \psi) \leftrightarrow (P\varphi \vee P\psi)$
- $F(\varphi \vee \psi) \leftrightarrow (F\varphi \wedge F\psi)$
- $\neg(O\varphi \wedge O\neg\varphi)$

The first and the last of these properties together with modus ponens, necessitation ($\varphi/O\varphi$) and a sufficient number of axioms of propositional logic can be used to axiomatize **SDL**. (This system coincides with the system **KD** in the classification of Chellas ([17]).)

Again the question arises whether the above properties are adequate for deontic reasoning. **SDL** suffers from a number of paradoxes, again mostly inherent in the (normal) modal semantics of the operators. For example, Ross's paradox: $O\varphi \rightarrow O(\varphi \vee \psi)$: if one ought to mail the letter then one ought to mail it or burn it. This sounds peculiar, but if one interprets O as holding in ideal alternative worlds it is evidently true. What is problematic here, is that in natural language an obligation of a disjunction is normally held to be an obligation of one of the disjuncts that may be chosen arbitrarily by the agent, and this intuition is simply not captured by SDL semantics. There are also more serious paradoxes, notably those having to do with contrary-to-duty (CTD) imperatives, in which certain obligations are specified in case one is already violating another obligation. For example, one ought to refrain from killing animals. But if one kills an animal, one ought to do it gently. This kind of CTD obligations cannot be expressed adequately in **SDL**. To reason about CTDs, or more generally about *conditional* obligations of the form $O(\varphi/\psi)$, read as the obligation to φ under circumstance ψ, so-called *dyadic deontic logic* was introduced, already in the 60s by von Wright [88]. However, in the 90s it became apparent that a truly adequate treatment of CTDs seems to force one to enter the realm of *nonmonotonic / defeasible / preferential* reasoning, which is beyond the scope of this chapter. More about this can be found in [58] and particularly [60, 15].

4.2 Dynamic deontic logic

Another issue that plays a role in deontic logic is the confusion about the argument of the modal operators. In **SDL** these are propositions (and we may refer to the O-operator as being of an 'ought-to be' nature). But many examples in the literature (and indeed also the example we gave illustrating Ross's paradox) actually seem to concern actions rather than propositions. (This is already noted by e.g. Castañeda [16].) One may also try and capture this notion of 'ought-to-do' in a different logic, and this is what we do next. **DDL**, introduced in [59], is a version of dynamic logic especially tuned to use as ought-to-do style deontic logic. It is based on the idea of Anderson's reduction of ought-to-be style deontic logic to alethic modal logic ([3]), but instead it reduces ought-to-do deontic logic to dynamic logic ([42]). The basic idea is very simple: some action is forbidden if doing the action leads to a state of violation. In a formula: $\hat{F}\alpha \leftrightarrow_{def} [\alpha]V$, where the dynamic logic formula $[\alpha]\varphi$ denotes that execution / performance of the action α leads (necessarily) to a state (or states) where φ holds, and V is a special atomic formula denoting violation. (We write $\hat{F}$ instead of F to indicate that this operator is of a different (viz. 'to-do') kind than the SDL operator; likewise for the other operators in this section.) Formally, we say that the meaning of action α is captured by an accessibility relation $R_\alpha \subseteq W \times W$ associated with α, where W is the set of possible worlds. This relation R_α describes exactly what possible moves (state transitions) are induced by performance of the action α: $R_\alpha(u, w)$ says that from world u one can get into world w by performing α. (In concurrency semantics and process algebra this is often specified by a so-called (labeled) transition system which enables one to derive (all) transitions of the kind $u \rightarrow_\alpha w$, which in fact defines the relation R_α for all possible actions α.) Now the formal meaning of the formula $[\alpha]\varphi$ is given by: $[\alpha]\varphi$ is true in a possible world w iff all states w' with

$R_\alpha(w, w')$ satisfy φ. This then provides the formal definition of the $\hat{F}$-operator, as given above. In the sequel we will also employ the dual $\langle\alpha\rangle$ of $[\alpha]$: $\langle\alpha\rangle\varphi$ is true in w iff there is some state w' satisfying φ such that $R_\alpha(w, w')$.

The other deontic modalities are derivatives of $\hat{F}$: permission is not-forbidden ($\hat{P}\alpha \leftrightarrow \neg\hat{F}\alpha$), and obligation is forbidden-not-to ($\hat{O}\alpha \leftrightarrow \hat{F}\bar{\alpha}$), where $\bar{\alpha}$ has the meaning of "not-α". The formal semantics of this negated action is non-trivial, especially in case one considers composite actions (cf. [59, 91, 26, 92]). In the cited papers we considered connectives for composing non-atomic actions, such as '$\cup$' (choice, the dynamic analogue of disjunction in a static setting), '&' (parallel, the analogue of conjunction), '$-$' (non-performance, the analogue of negation), and ';' (sequential composition, which has no analogue in a static setting). Without giving a formal semantics here (see the papers mentioned above for that), the meaning of these are as follows: $\alpha_1 \cup \alpha_2$ expresses a choice between α_1 and α_2 (this—roughly—corresponds to taking $R_{\alpha_1 \cup \alpha_2}$ as the set-theoretic union of R_{α_1} and R_{α_2}), $\alpha_1 \& \alpha_2$ a parallel performance of α_1 and α_2 (this amounts to more or less taking $R_{\alpha_1 \& \alpha_2}$ to be the intersection of R_{α_1} and R_{α_2}), $\bar{\alpha}$ (we will also write $-\alpha$) the non-performance of α, as stated above (it more or less amounts to taking $R_{\bar{\alpha}}$ to be some complement of R_α, but see also the discussion below), and $\alpha_1; \alpha_2$ the performance of α_1 followed by that of α_2. For a full account of the semantics of particularly negated actions we refer to [59, 26, 23, 25].

With this semantics the following formulas are valid:

PROPOSITION 8.

- $[\alpha](\varphi \to \psi) \to ([\alpha]\varphi \to [\alpha]\psi)$
- $[\alpha; \beta]\varphi \leftrightarrow [\alpha][\beta]\varphi$
- $[\alpha \cup \beta]\varphi \leftrightarrow ([\alpha]\varphi \wedge [\beta]\varphi)$
- $[\alpha]\varphi \to [\alpha\&\beta]\varphi$
- $[-(\alpha; \beta)]\varphi \leftrightarrow ([-\alpha]\varphi \wedge [\alpha][-\beta]\varphi)$
- $[-\alpha]\varphi \to [-(\alpha \cup \beta)]\varphi$
- $[-(\alpha\&\beta)]\varphi \leftrightarrow ([-\alpha]\varphi \wedge [-\beta]\varphi)$
- $\hat{F}\alpha \leftrightarrow [\alpha]V$
- $\hat{P}\alpha \leftrightarrow \neg\hat{F}\alpha (\leftrightarrow \langle\alpha\rangle\neg V)$
- $\hat{O}\alpha \leftrightarrow \hat{F}(-\alpha)(\leftrightarrow [-\alpha]V)$
- $\hat{O}(\alpha; \beta) \leftrightarrow (\hat{O}\alpha \wedge [\alpha]\hat{O}\beta)$
- $\hat{P}(\alpha; \beta) \leftrightarrow \langle\alpha\rangle\hat{P}\beta$
- $\hat{F}(\alpha; \beta) \leftrightarrow [\alpha]\hat{F}\beta$
- $\hat{O}(\alpha\&\beta) \leftrightarrow (\hat{O}\alpha \wedge \hat{O}\beta)$
- $\hat{P}(\alpha\&\beta) \to (\hat{P}\alpha \wedge \hat{P}\beta)$

- $(\hat{F}\alpha \vee \hat{F}\beta) \rightarrow \hat{F}(\alpha \& \beta)$
- $(\hat{O}\alpha \vee \hat{O}\beta) \rightarrow \hat{O}(\alpha \cup \beta)$
- $\hat{P}(\alpha \cup \beta) \leftrightarrow (\hat{P}\alpha \vee \hat{P}\beta)$
- $\hat{F}(\alpha \cup \beta) \leftrightarrow (\hat{F}\alpha \wedge \hat{F}\beta)$

Modal action logics that contain action negation (complement) operators have been studied by several authors, for instance [20, 37, 9]. From these studies, in particular [9], it has become clear that there are several ways to define action negation, particularly in the context of the operators ';' and intersection. The choices made in **DDL** above were motivated mainly by the desirability of the validities concerning the deontic operators above. There have been proposed several dynamic deontic logics that could be viewed as some kind of refinement of the original logic as presented here. These concern -amongst other ones- issues of context (pertaining to the kind of complement / negation again) [23], the exact way action bring about violations [24], and of a more refined view of action than just input-output relations [78, 13].

5 BDI LOGIC

BDI logic as proposed by Rao & Georgeff [62] came about after the ground-breaking work of Bratman [8] on the philosophy of intelligent (human) agents. In this work Bratman made a case for the notion of *intention* besides belief and desire, to describe the behavior of rational agents. Intentions force the agent to commit to certain desires and to really 'go for them'. So focus of attention is an important aspect here, which also enables the agent to monitor how s/he is doing and take measures if things go wrong. Rao & Georgeff stress that in the case of resource-bounded agents it is imperative to focus on desires / goals and make choices. This was also observed by Cohen & Levesque [18], who tried to formalize the notion of intention in a linear-time temporal logic in terms of the notion of a (persistent) goal. Here we follow Rao & Georgeff who use a branching-time temporal logic framework to give a formal-logical account of BDI theory. BDI logic has influenced many researchers (including Rao & Georgeff themselves) to think about architectures of agent-based systems in order to realize these systems. Rao & Georgeff's BDI logic is more liberal than that of Cohen & Levesque in the sense that they *a priori* regard each of the three attitudes of belief, desire and intention as primitive: they introduce separate modal operators for belief, desire and intention, and then study possible relations between them.

The language of BDI logic is defined as follows. Two types of formulas are distinguished: state formulas and path formulas. We assume some given first-order signature. Furthermore, we assume a set E of event types with typical element e. The operators $BEL, GOAL, INTEND$ have as obvious intended reading the belief, goal and intention of an agent, respectively, while $\mathsf{U}, \Diamond, \mathsf{O}$ are the usual temporal operators, viz. 'until', 'eventually' and 'next', respectively.

DEFINITION 9. (State and path formulas.)

1. The set of *state formulas* is the smallest closed under the following conditions:
    - any first-order formula with respect to the given signature is a state formula;

- if φ_1 and φ_2 are state formulas then also $\neg\varphi_1, \varphi_1 \vee \varphi_2, \exists x \varphi_1$ are state formulas;
- if e is an event type, then $succeeded(e), failed(e)$ are state formulas;
- if φ is a state formula, then $BEL(\varphi), GOAL(\varphi), INTEND(\varphi)$ are state formulas;
- if ψ is a *path formula*, then $optional(\psi)$ is a state formula.

2. The set of *path formulas* is the smallest set closed under:
   - any state formula is a path formula;
   - if ψ_1, ψ_2 are path formulas, then $\neg\psi_1, \psi_1 \vee \psi_2, \psi_1 \mathsf{U} \psi_2, \Diamond\psi_1, \mathsf{O}\psi_1$ are path formulas.

State formulas are interpreted over a state, that is a (state of the) world at a particular point in time, while path formulas are interpreted over a path of a time tree (representing the evolution of a world). In the sequel we will see how this will be done formally. Here we just give the informal readings of the operators. The operators *succeeded* and *failed* are used to express that events have (just) succeeded and failed, respectively. As in the framework of Cohen & Levesque action-like entities should be given a place in the theory by means of additional operators. Here we see that Rao & Georgeff's approach also account for the distinction of trying an action / event and succeeding versus failing. With the latter one may think of several things: either the agent tried to do some action which failed due to circumstances in the environment. For example, for an action 'grip' to be successful there should be an object to be gripped; for a motor to be started there should be fuel, etc.; perhaps there is also some internal capacity missing needed for successful performance of an action: again for an action 'grip' to be successful the robot should have a gripper. This is related to the well-known *qualification problem in AI*, [67].

Next there are the modal operators for belief, goal and intend. (In the original version of BDI theory [62], desires are represented by goals, or rather a GOAL operator. In a later paper [64] the *GOAL* operator was replaced by *DES* for desire.) The optional operator states that there is a future (represented by a path) where the argument of the operator holds. Finally, there are the familiar (linear-time) temporal operators, such as the 'until', 'eventually' and 'next time', which are to be interpreted along a linear time path.

Furthermore, the following abbreviations are defined:

DEFINITION 10.

1. $\Box\psi = \neg\Diamond\neg\psi$ (always);
2. $inevitable(\psi) = \neg optional(\neg\psi)$;
3. $done(e) = succeeded(e) \vee failed(e)$;
4. $succeeds(e) = inevitable\mathsf{O}(succeeded(e))$;
5. $fails(e) = inevitable\mathsf{O}(failed(e))$;
6. $does(e) = inevitable\mathsf{O}(done(e))$.

The 'always' operator is the familiar one from (linear-time) temporal logic. The 'inevitability' operator expresses that its argument holds along all possible futures (paths from the current time). The 'done' operator states that an event occurs (action is done) no matter whether it is succeeding or not. The final three operators state that an event succeeds, fails, or is done iff it is inevitable (i.e. in any possible future) it is the case that at the next instance the event has succeeded, failed, or has been done, respectively. (So, this means that an event, succeeding or failing, is supposed to take one unit of time!)

DEFINITION 11. (Semantics)
The semantics is given with respect to models of the form $\mathfrak{M} = \langle W, E, T, \prec, \mathcal{U}, B, G, I, \Phi\rangle$, where

- W is a set of possible worlds;
- E is a set of primitive event types;
- T is a set of time points;
- $\prec$ is a binary relation on time points, which is serial, transitive and back-wards linear;
- $\mathcal{U}$ is the universe of discourse;
- Φ is a mapping of first-order entities to $\mathcal{U}$, for any world and time point;
- $B, G, I \subseteq W \times T \times W$ are accessibility relations for $BEL, GOAL, INTEND$, respectively.

The semantics of BDI logic, Rao & Georgeff-style, is rather complicated. Of course, we have possible worlds again, but as we will see below, these are not just unstructured elements, but they are each time trees, describing possible flows of time. So, we also need time points and an ordering on them. As BDI logic is based on branching time, the ordering need not be linear in the sense that all time points are related in this ordering. However, it is stipulated that the time ordering is serial (every time point has a successor in the time ordering), the ordering is transitive and backwards-linear, which means that every time point has only one direct predecessor. The accessibility relations for the 'BDI'-modalities are standard apart from the fact that they are also time-related, that is to say that worlds are (belief/goal/intend-)accessible with respect to a time point. Another way of viewing this is that – for all three modalities – for every time point there is a distinct accessibility relation between worlds.

Next we elaborate on the structure of the possible worlds.

DEFINITION 12. (Possible worlds)
Possible worlds in W are assumed to be *time trees*: an element $w \in W$ has the form $w = \langle T_w, A_w, S_w, F_w\rangle$, where

- $T_w \subseteq T$ is the set of time points in world w;
- A_w is the restriction of the relation $\prec$ to T_w;
- $S_w : T_w \times T_w \to E$ maps adjacent time points to (successful) events;

- $F_w : T_w \times T_w \to E$ maps adjacent time points to (failing) events;
- the domains of the functions S_w and F_w are disjoint.

As announced before, a possible world itself is a time tree, a temporal structure representing possible flows of time. The definition above is just a technical one stating that the time relation within a possible world derives naturally from the *a priori* given relation on time points. Furthermore it is indicated by means of the functions S_w and F_w how events are associated with adjacent time points.

Now we come to the formal interpretation of formulas on the above models. Naturally we distinguish state formulas and path formulas, since the former should be interpreted on states whereas the latter are interpreted on paths. In the sequel we use the notion of a *fullpath*: a fullpath in a world w is an *infinite* sequence of time points such that, for all i, $(t_i, t_{i+1}) \in A_w$. We denote a fullpath in w by $(w_{t0}, w_{t1}, \ldots)$, and define $fullpaths(w)$ as the set of all fullpaths occurring in world w (i.e. all fullpaths that start somewhere in the time tree w).

DEFINITION 13. (Interpretation of formulas)
The interpretation of formulas with respect to a model

$$\mathfrak{M} = \langle W, E, T, \prec, \mathcal{U}, B, G, I, \Phi \rangle$$

is given by:

1. (state formulas)

   - $\mathfrak{M}, v, w_t \models q(y_1, \ldots, y_n) \leftrightarrow (v(y_1), \ldots, v(y_n)) \in \Phi(q, w, t)$;
   - $\mathfrak{M}, v, w_t \models \neg\varphi \leftrightarrow \mathfrak{M}, v, w_t \not\models \varphi$;
   - $\mathfrak{M}, v, w_t \models \varphi_1 \vee \varphi_2 \leftrightarrow \mathfrak{M}, v, w_t \models \varphi_1$ or $\mathfrak{M}, v, w_t \models \varphi_2$;
   - $\mathfrak{M}, v, w_t \models \exists x \varphi \leftrightarrow \mathfrak{M}, v\{d/x\}, w_t \models \varphi$ for some $d \in \mathcal{U}$;
   - $\mathfrak{M}, v, w_{t0} \models optional(\psi) \leftrightarrow$ there exists some fullpath $(w_{t0}, w_{t1}, \ldots)$ such that $\mathfrak{M}, v, (w_{t0}, w_{t1}, \ldots) \models \psi$;
   - $\mathfrak{M}, v, w_t \models BEL(\varphi) \leftrightarrow$ for all $w' \in B(w,t) : \mathfrak{M}, v, w'_t \models \varphi$;
   - $\mathfrak{M}, v, w_t \models GOAL(\varphi) \leftrightarrow$ for all $w' \in G(w,t) : \mathfrak{M}, v, w'_t \models \varphi$;
   - $\mathfrak{M}, v, w_t \models INTEND(\varphi) \leftrightarrow$ for all $w' \in I(w,t) : \mathfrak{M}, v, w'_t \models \varphi$;
   - $\mathfrak{M}, v, w_t \models succeeded(e) \leftrightarrow$ exists $t0$ such that $S_w(t0, t) = e$;
   - $\mathfrak{M}, v, w_t \models failed(e) \leftrightarrow$ exists $t0$ such that $F_w(t0, t) = e$.

   In the above and elsewhere $v\{d/x\}$ denotes the function v modified such that $v(x) = d$, and $R(w,t) = \{w' \mid R(w,t,w')\}$ for $R = B, G, I$

2. (path formulas)

   - $\mathfrak{M}, v, (w_{t0}, w_{t1}, \ldots) \models \varphi \leftrightarrow \mathfrak{M}, v, w_{t0} \models \varphi$, for φ state formula;
   - $\mathfrak{M}, v, (w_{t0}, w_{t1}, \ldots) \models \mathsf{O}\varphi \leftrightarrow \mathfrak{M}, v, (w_{t1}, w_{t2}, \ldots) \models \varphi$;
   - $\mathfrak{M}, v, (w_{t0}, w_{t1}, \ldots) \models \Diamond\varphi \leftrightarrow \mathfrak{M}, v, (w_{tk}, \ldots) \models \varphi$ for some $k \geq 0$;

- $\mathfrak{M}, v, (w_{t0}, w_{t1}, \ldots) \models \psi_1 \mathsf{U} \psi_2 \leftrightarrow$ either (i) there exists $k \geq 0$ such that $\mathfrak{M}, v, (w_{tk}, \ldots) \models \psi_2$ and for all $0 \leq j < k$: $\mathfrak{M}, v, (w_{tj}, \ldots) \models \psi_1$, or (ii) for all $j \geq 0$: $\mathfrak{M}, v, (w_{tj}, \ldots) \models \psi_1$.

Most of the above clauses should be clear, including those concerning the modal operators for belief, goal and intention. The clause for the 'optional' operator expresses exactly that optionally ψ is true if ψ is true in one of the possible futures represented by fullpaths starting at the present time point. The interpretation of the temporal operators is as usual.

Rao & Georgeff now discuss a number of properties that may be desirable to have as axioms. In the following we use α to denote so-called *O-formulas*, which are formulas that contain no positive occurrences of the '*inevitable*' operator (or negative occurrences of '*optional*") outside the scope of the modal operators $BEL, GOAL$ and $INTEND$.

1. $GOAL(\alpha) \rightarrow BEL(\alpha)$
2. $INTEND(\alpha) \rightarrow GOAL(\alpha)$
3. $INTEND(does(e)) \rightarrow does(e)$
4. $INTEND(\varphi) \rightarrow BEL(INTEND(\varphi))$
5. $GOAL(\varphi) \rightarrow BEL(GOAL(\varphi))$
6. $INTEND(\varphi) \rightarrow GOAL(INTEND(\varphi))$
7. $done(e) \rightarrow BEL(done(e))$;
8. $INTEND(\varphi) \rightarrow inevitable\Diamond(\neg INTEND(\varphi))$

In order to render these formulas validities further constraints should be imposed on the models, since in the general setting above these are not yet valid. For reasons of space we only consider the first two. (More can be found in [62, 64, 93].) In order to define constraints on the models such that these two become valid, we introduce the relation $\lhd$ on worlds, as follows: $w'' \lhd w' \Leftrightarrow fullpaths(w'') \subseteq fullpaths(w')$. So, $w'' \lhd w'$ means that there the world (time tree) w'' represents less choices than w'. Now we define the *B-G condition* as the property that the following holds:

$$\forall w' \in B(w, t)\ \exists w'' \in G(w, t) : w'' \lhd w'$$

Informally, this condition says that for any belief accessible world there is a goal accessible world that contains less choices. It is now easy to prove the following proposition.

PROPOSITION 14.
Let $\mathcal{BG}$ be the class of models of the above form that satisfy the B-G condition. Then

$$\mathcal{BG} \models GOAL(\alpha) \rightarrow BEL(\alpha)$$

for O-formulas α.

Similarly one can define the *G-I condition* as

$$\forall w' \in G(w, t)\ \exists w'' \in I(w, t) : w'' \lhd w'$$

and obtain:

PROPOSITION 15.
Let $\mathcal{GI}$ be the class of models of the above form that satisfy the G-I condition. Then

$$\mathcal{GI} \models INTEND(\alpha) \rightarrow GOAL(\alpha)$$

for O-formulas α.

Let us now consider the properties deemed desirable by Rao & Georgeff again. Actually, the first one is rather controversial. (Cohen & Levesque had the inverse implication in their framework, although admittedly that framework is quite different from Rao & Georgeff's because of the different temporal model – linear time instead of branching time, so that it is not completely fair to compare formulas...!) Rao & Georgeff try to render the formula concerned (which they call 'belief-goal compatibility') plausible by considering a typical O-formula α of the form $optional(\psi)$, and then note that if it is a goal that something is optional (true in some future) then it should also be believed that it is optional (true in some future). This, indeed, sounds plausible in the sense that a rational and realistic agent would adhere to it. But also objective (nonmodal) formulas are O-formulas, and whether this property is also plausible for these formulas is debatable.

The second formula is similar to the first. This one is called goal-intention compatibility, and is defended by Rao & Georgeff by stating that if an optionality is intended it should also be wished (a goal in their terms). So, Rao & Georgeff have a kind of selection filter in mind: intentions (or rather intended options) are filtered / selected goals (or rather goal (wished) options), and goal options are selected believed options.

The third one says that the agent really does the primitive actions that s/he intends to do. This means that if one adopts this as an axiom the agent is not allowed to do something else (first).

The fourth, fifth and seventh express that the agent is conscious of its intentions, goals and what primitive action he has done in the sense that he believes what he intends, has as a goal and what primitive action he has just done. The sixth one says something like that intentions are really wished for: if something is an intention then it is a goal that it is an intention.

The eighth formula states that intentions will inevitably (in every possible future) be dropped eventually, so there is no infinite deferral of its intentions. This leaves open, whether the intention will be fulfilled eventually, or will be given up for other reasons. Below we will discuss several possibilities of giving up intentions according to different types of commitment an agent may have.

BDI-logical expressions can be used to characterize different types of agents. Rao & Georgeff mention the following possibilities:

1. (blindly committed agent) $INTEND(inevitable\Diamond\varphi) \rightarrow$
   $inevitable(INTEND(inevitable\Diamond\varphi)\mathsf{U}BEL(\varphi))$

2. (single-minded committed agent) $INTEND(inevitable\Diamond\varphi) \rightarrow$
   $inevitable(INTEND(inevitable\Diamond\varphi)\mathsf{U}(BEL(\varphi) \vee \neg BEL(optional\Diamond\varphi)))$

3. (open minded committed agent) $INTEND(inevitable\Diamond\varphi) \rightarrow$
   $inevitable(INTEND(inevitable\Diamond\varphi)\mathsf{U}(BEL(\varphi) \vee \neg GOAL(optional\Diamond\varphi)))$

A blindly committed agent maintains his intentions to inevitably obtaining eventually something until he actually believes that that something has been fulfilled. A single-minded committed agent is somewhat more flexible: he maintains his intention until he believes he has achieved it *or he does not believe that it can be reached (it is still an option in some future) anymore*. Finally, the open minded committed agent is even more flexible: he can also drop his intention if it is not a goal (desire) anymore.

Rao & Georgeff obtain results under which conditions the various types of committed agents will reach their intentions. For example, under the assumption of the axioms we have discussed earlier it holds for a blindly committed agent that:

$$INTEND(inevitable\Diamond\varphi) \to inevitable\Diamond BEL(\varphi)$$

expressing that if the agent intends to eventually obtain φ (s)he will inevitably eventually believe that it has succeeded in achieving φ.

6 KARO LOGIC

In this section we turn to the KARO formalism, in which *action* rather than time, together with knowledge / belief, is the primary concept, on which other agent notions are built. The KARO framework has been developed in a number of papers (e.g. [80, 81, 77, 57]) as well as in the thesis of Van Linder ([79]).

The KARO formalism is an amalgam of dynamic logic and epistemic / doxastic logic, augmented with several additional (modal) operators in order to deal with the motivational aspects of agents. So, besides operators for knowledge (**K**), belief (**B**) and action ($[\alpha]$, "after performance of α it holds that"), there are additional operators for ability (**A**) and desire (**D**).

Assume a set $\mathcal{A}$ of atomic actions and a set $\mathcal{P}$ of atomic propositions.

DEFINITION 16. (Language)
The language $\mathcal{L}_{KARO}$ of KARO-formulas is given by the BNF grammar:

$$\begin{array}{rcl} \varphi & ::= & p(\in \mathcal{P}) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \ldots \\ & & \mathbf{K}\varphi \mid \mathbf{B}\varphi \mid \mathbf{D}\varphi \mid [\alpha]\varphi \mid \mathbf{A}\alpha \end{array}$$

$$\begin{array}{rcl} \alpha & ::= & a(\in \mathcal{A}) \mid \alpha_1;\alpha_2 \mid \varphi? \mid \\ & & \texttt{if}\ \varphi\ \texttt{then}\ \alpha_1\ \texttt{else}\ \alpha_2\ \texttt{fi} \mid \\ & & \texttt{while}\ \varphi\ \texttt{do}\ \alpha\ \texttt{od} \end{array}$$

Thus formulas are built by means of the familiar propositional connectives and the modal operators for knowledge, belief, desire, action and ability. The action expressions defined in the second (α)-clause are familiar from imperative programming: atomic actions, tests and sequential composition, conditional and repetition.

DEFINITION 17. (KARO models)

1. The semantics of the knowledge, belief and desires operators is given by means of Kripke structures of the following form: $\mathfrak{M} = \langle W, V, R_K, R_B, R_D \rangle$, where
   - W is a non-empty set of states (or worlds);
   - V is a truth assignment function per state;
   - R_K, R_B, R_D are accessibility relations for interpreting the modal operators $\mathbf{K}, \mathbf{B}, \mathbf{D}$.

     The relation R_K is assumed to be an equivalence relation, while the relation R_B is assumed to be euclidean, transitive and serial. Furthermore we assume that $R_B \subseteq R_K$. (No special constraints are assumed for the relations R_D.)
2. The semantics of actions is given by means of structures of type $\langle \Sigma, \{R_a \mid a \in \mathcal{A}\}, \mathcal{C}, Ag \rangle$, where
   - Σ is the set of possible model/state pairs (i.e. models of the above form, together with a state appearing in that model);
   - R_a ($a \in \mathcal{A}$) are relations on Σ encoding the behavior of atomic actions;
   - $\mathcal{C}$ is a function that gives the set of actions that the agent is able to do per model/state pair;
   - Ag is a function that yields the set of actions that the agent is committed to (the agent's 'agenda') per model/state pair.

Knowledge, belief, and desire are modeled by accessibility relationson worlds, as usual. Actions are modelled as model/state pair transformers to emphasize their influence on the mental state (that is, the complex of knowledge, belief and desires) of the agent rather than just the state of the world. Both (cap)abilities and commitments are given by functions that yield the relevant information per model / state pair.

DEFINITION 18. (Interpretation of formulas)
In order to determine whether a formula $\varphi \in \mathcal{L}$ is true in a model/state pair $(\mathfrak{M}, w)$, we stipulate:

- $\mathfrak{M}, w \models p$ iff $V(w)(p) = \mathit{true}$, for $p \in \mathcal{P}$;
- the logical connectives are interpreted as usual;
- $\mathfrak{M}, w \models \mathbf{K}\varphi$ iff $\mathfrak{M}, w' \models \varphi$ for all w' with $R_K(w, w')$;
- $\mathfrak{M}, w \models \mathbf{B}\varphi$ iff $\mathfrak{M}, w' \models \varphi$ for all w' with $R_B(w, w')$;
- $\mathfrak{M}, w \models \mathbf{D}\varphi$ iff $\mathfrak{M}, w' \models \varphi$ for all w' with $R_D(w, w')$;
- $\mathfrak{M}, w \models [\alpha]\varphi$ iff $\mathfrak{M}', w' \models \varphi$ for all $\mathfrak{M}', w'$ with $R_\alpha((\mathfrak{M}, w), (\mathfrak{M}', w'))$;
- $\mathfrak{M}, w \models \mathbf{A}\alpha$ iff $\alpha \in \mathcal{C}(\mathfrak{M}, w)$;[2]

[2] In [76] we have shown that the ability operator can alternatively defined by means of a second accessibility relation for actions, in a way analogous to the opportunity operator below.

- $\mathfrak{M}, w \models \mathbf{Com}(\alpha)$ iff $\alpha \in Ag(\mathfrak{M}, w)$.[3]

Here R_α is defined as usual in dynamic logic by induction from the basic case R_a (cf. e.g. [43, 79, 77], but now on model/state pairs rather than just states). Likewise the function $\mathcal{C}$ is lifted to sets of complex actions ([79, 77]).

Knowledge, belief and desire are interpreted as modal operators, as usual. The action modality gets a similar interpretation: something (necessarily) holds after the performance / execution of action α if it holds in all the situations that are accessible from the current one by doing the action α. The only thing which is slightly nonstandard is that a situation is characterized here as a model / state pair. The interpretations of the ability and commitment operators are rather trivial in this setting (but see the footnotes): an action is enabled (or rather: the agent is able to do the action) if it is indicated so by the function C, and, likewise, an agent is committed to an action α if it is recorded so in the agent's agenda.

Furthermore, we will make use of the following syntactic abbreviations serving as auxiliary operators:

DEFINITION 19.

- (dual) $\langle\alpha\rangle\varphi = \neg[\alpha]\neg\varphi$, expressing that the agent has the opportunity to perform α resulting in a state where φ holds.
- (opportunity) $\mathbf{O}\alpha = \langle\alpha\rangle\top$, i.e., an agent has the opportunity to do an action iff there is a successor state with respect to the R_α-relation;
- (practical possibility) $\mathbf{P}(\alpha, \varphi) = \mathbf{A}\alpha \wedge \mathbf{O}\alpha \wedge \langle\alpha\rangle\varphi$, i.e., an agent has the practical possibility to do an action with result φ iff it is both able and has the opportunity to do that action and the result of actually doing that action leads to a state where φ holds;
- (can) $\mathbf{Can}(\alpha, \varphi) = \mathbf{KP}(\alpha, \varphi)$, i.e., an agent can do an action with a certain result iff it knows it has the practical possibility to do so;
- (realizability) $\Diamond\varphi = \exists a_1, \ldots, a_n \mathbf{P}(a_1; \ldots; a_n, \varphi)$[4], i.e., a state property φ is realizable iff there is a finite sequence of atomic actions of which the agent has the practical possibility to perform it with the result φ;
- (goal) $\mathbf{G}\varphi = \neg\varphi \wedge \mathbf{D}\varphi \wedge \Diamond\varphi$, i.e., a goal is a formula that is not (yet) satisfied, but desired and realizable.[5]
- (possible intend) $\mathbf{I}(\alpha, \varphi) = \mathbf{Can}(\alpha, \varphi) \wedge \mathbf{KG}\varphi$, i.e., an agent (possibly) intends an action with a certain result iff the agent can do the action with that result and it moreover knows that this result is one of its goals.

[3] The agenda is assumed to be closed under certain conditions such as taking 'prefixes' of actions (representing initial computations). Details are omitted here, but can be found in [57].

[4] We abuse our language here slightly, since strictly speaking we do not have quantification in our object language. See [57] for a proper definition.

[5] In fact, here we simplify matters slightly. In [57] we also stipulate that a goal should be explicitly selected somehow from the desires it has, which is modelled in that paper by means of an additional modal operator. Here we leave this out for simplicity's sake.

REMARK

- The dual of the (box-type) action modality expresses that there is at least a resulting state where a formula φ holds. It is important to note that in the context of *deterministic* actions, i.e. actions that have at most one successor state, this means that the *only* state satisfies φ, and is thus in this particular case a stronger assertion than its dual formula $[\alpha]\varphi$, which merely states that if there are any successor states they will (all) satisfy φ. Note also that if atomic actions are assumed to be deterministic all actions including the complex ones will be deterministic.

- Opportunity to do an action is modelled by having at least one successor state according to the accessibility relation associated with the action.

- Practical possibility to to an action with a certain result is modelled as having both ability and opportunity to do the action with the appropriate result. Note that $\mathbf{O}\alpha$ in the formula $\mathbf{A}\alpha \wedge \mathbf{O}\alpha \wedge \langle\alpha\rangle\varphi$ is actually redundant since it already follows from $\langle\alpha\rangle\varphi$. However, to stress the opportunity aspect it is added.

- The Can predicate applied to an action and formula expresses that the agent is 'conscious' of its practical possibility to do the action resulting in a state where the formula holds.

- A formula φ is realizable if there is a 'plan' consisting of (a sequence of) atomic actions of which the agent has the practical possibility to do them with φ as a result.

- A formula φ is a goal in the KARO framework if it is not true yet, but desired and realizable in the above meaning, that is, there is a plan of which the agent has the practical possibility to realize it with φ as a result.

- An agent is said to (possibly) intend an action α with result φ if he Can do this (knows that he has the practical possibility to do so), and, moreover, knows that φ is a goal.

In order to manipulate both knowledge / belief and motivational matters special actions `revise`, `commit` and `uncommit` are added to the language. These operators cannot be nested. So, e.g., $\texttt{commit}(\texttt{uncommit}\alpha)$ is not a well-formed action expression. (For a proper definition of the language the reader is referred to [57].) The semantics of these actions are again given as model/state transformers (We only do this here in a very abstract manner, viewing the accessibility relations associated with these actions as functions. For further details we refer to e.g. [79, 77, 57]):

DEFINITION 20. (Accessibility of revise, commit and uncommit actions)

1. $R_{\texttt{revise}\varphi}(\mathfrak{M}, w) = \mathit{update\_belief}(\varphi, (\mathfrak{M}, w))$.

2. $R_{\texttt{commit}\alpha}(\mathfrak{M}, w) = \mathit{update\_agenda}^{+}(\alpha, (\mathfrak{M}, w))$, if $\mathfrak{M}, w \models \mathbf{I}(\alpha, \varphi)$ for some φ, otherwise $R_{\texttt{commit}\alpha}(\mathfrak{M}, w) = \emptyset$ (indicating failure of the commit action).

3. $R_{\texttt{uncommit}\alpha}(\mathfrak{M}, w) = \mathit{update\_agenda}^{-}(\alpha, (\mathfrak{M}, w))$, if $\mathfrak{M}, w \models \mathbf{Com}(\alpha)$, otherwise $R_{\texttt{uncommit}\alpha}(\mathfrak{M}, w) = \emptyset$ (indicating failure of the uncommit action).

4. $\texttt{uncommit}\alpha \in \mathcal{C}(\mathfrak{M}, w)$ iff $\mathfrak{M}, w \models \neg\mathbf{I}(\alpha, \varphi)$ for all formulas φ, that is, an agent is able to uncommit to an action if it is not intended to do it (any longer) for any purpose.

Here *update_belief*, *update_agenda*$^+$ and *update_agenda*$^-$ are functions that update the agent's belief and agenda (by adding or removing an action), respectively. Details are omitted here, but essentially these actions are model/state transformers again, representing a change of the mental state of the agent (regarding beliefs and commitments, respectively). The *update_belief*$(\varphi, (\mathfrak{M}, w))$ function changes the model $\mathfrak{M}$ in such a way that the agent's belief is updated with the formula φ, while *update_agenda*$^+(\alpha, (\mathfrak{M}, w))$ changes the model $\mathfrak{M}$ such that α is added to the agenda. The same holds for the *update_agenda*$^-$ function, but now with respect to removing an action from the agenda. The formal definitions can be found in [80, 81] and [57]. The `revise` operator can be used to cater for revisions due to observations and communication with other agents, which we will not go into further here (see [81]).

The interpretation of formulas containing revise and (un)commit actions is now done using the accessibility relations above. One can now define validity as usual with respect to the KARO-models. One then obtains the following validities (of course, in order to be able to verify these one should use the proper model and not the abstraction we have presented here.) Besides the familiar properties from epistemic / doxastic logic, typical properties of this framework, called the KARO logic, include (cf. [80, 57]):

PROPOSITION 21.

1. $\models \mathbf{O}(\alpha;\beta) \leftrightarrow \langle\alpha\rangle\mathbf{O}\beta$
2. $\models \mathbf{Can}(\alpha;\beta,\varphi) \leftrightarrow \mathbf{Can}(\alpha, \mathbf{P}(\beta,\varphi))$
3. $\models [\texttt{revise}\varphi]\mathbf{B}\varphi$
4. $\models \mathbf{K}\neg\varphi \leftrightarrow [\texttt{revise}\varphi]\mathbf{B}\bot$
5. $\models \mathbf{K}(\varphi \leftrightarrow \psi) \rightarrow ([\texttt{revise}\varphi]\mathbf{B}\chi \leftrightarrow [\texttt{revise}\psi]\mathbf{B}\chi)$
6. $\models \mathbf{I}(\alpha,\varphi) \rightarrow \langle\texttt{commit}\alpha\rangle\mathbf{Com}(\alpha)$
7. $\models \mathbf{I}(\alpha,\varphi) \rightarrow \neg\mathbf{A}\texttt{uncommit}(\alpha)$
8. $\models \mathbf{Com}(\alpha) \rightarrow \langle\texttt{uncommit}(\alpha)\rangle\neg\mathbf{Com}(\alpha)$
9. $\models \mathbf{Com}(\alpha) \wedge \neg\mathbf{Can}(\alpha,\top) \rightarrow \mathbf{Can}(\texttt{uncommit}(\alpha), \neg\mathbf{Com}(\alpha))$
10. $\models \mathbf{Com}(\alpha) \rightarrow \mathbf{KCom}(\alpha)$
11. $\models \mathbf{Com}(\alpha_1;\alpha_2) \rightarrow \mathbf{Com}(\alpha_1) \wedge \mathbf{K}[\alpha_1]\mathbf{Com}(\alpha_2)$
12. $\models \mathbf{Com}(\texttt{if } \varphi \texttt{ then } \alpha_1 \texttt{ else } \alpha_2 \texttt{ fi}) \wedge \mathbf{K}\varphi \rightarrow \mathbf{Com}(\varphi?;\alpha_1)$
13. $\models \mathbf{Com}(\texttt{if } \varphi \texttt{ then } \alpha_1 \texttt{ else } \alpha_2 \texttt{ fi}) \wedge \mathbf{K}\neg\varphi \rightarrow \mathbf{Com}(\neg\varphi?;\alpha_2)$
14. $\models \mathbf{Com}(\texttt{while } \varphi \texttt{ do } \alpha \texttt{ od}) \wedge \mathbf{K}\varphi \rightarrow \mathbf{Com}((\varphi?;\alpha);\texttt{while } \varphi \texttt{ do } \alpha \texttt{ od})$

The first of these properties says that having the opportunity to do a sequential composition of two actions amounts to having the opportunity of doing the first action first and then having the opportunity to do the second. The second states that an agent that *can* do a sequential composition of two actions with result φ iff the agent can do the first actions resulting in a state where it has the practical possibility to do the second with φ as result. The third expresses that a revision with φ results in a belief of φ. The fourth states that the revision with φ results in inconsistent belief iff the agent knows $\neg\varphi$ for certain. The fifth expresses that revisions with formulas that are known to be equivalent have identical results. The sixth asserts that if an agent possibly intends to do α with some result φ, it has the opportunity to commit to α with result that it is committed to α (i.e. α is put into its agenda). The seventh says that if an agent intends to do α with a certain purpose, then it is unable to uncommit to it (so, if it is committed to α it has to persevere in it). The eighth property says that if an agent is committed to an action and it has the opportunity to uncommit to it with as result that indeed the commitment is removed. The ninth says that whenever an agent is committed to an action that is no longer known to be practically possible, it knows that it can undo this impossible commitment. The tenth property states that commitments are known to the agent. The last four properties have to do with commitments to complex actions. For instance, the eleventh says that if an agent is committed to a sequential composition of two actions then it is committed to the first one, and it knows that after doing the first action it will be committed to the second action.

The KARO framework has been extended in various ways. In [76] we have given an account of abilities based on dynamic logic (like we did already for results and opportunities). In [46] we considered *automated reasoning* (viz. resolution) methods for (a fragment of) KARO. Furthermore, Dignum and Van Linder [28] have extended it to deal with *speech acts*. Aldewereld *et al.* [2] have extended KARO with multi-agent notions such as *joint* beliefs, actions, goals and commitments. (We will return to this briefly in the next section.) Finally we mention that KARO can also be employed beyond the realm of rational agents: in [55] it is indicated how the framework may be used to describe *emotional* aspects of agency.

7 MULTI-AGENT LOGICS

In the previous sections we have concentrated mainly on single agents and how to describe them. Of course, if multiple agents are around, things become both more complicated as well as more interesting. In this section we will look at two generalizations of single-agent logics to multi-agent logics, viz. multi-agent epistemic logic and multi-agent BDI logic.

7.1 Multi-agent epistemic logic

In a multi-agent setting one can extend a single-agent framework in several ways. To start with, with respect to the epistemic (doxastic) aspect, one can introduce epistemic (doxastic) operators for every agent, resulting in a multi-modal logic, called $\mathbf{S5}_n$. Models for this logic are inherently less simple and elegant as those for the single agent case (cf. [56]). One has indexed operators $\mathbf{K}_i$ and $\mathbf{B}_i$ for agent i's knowledge and belief, respectively. But one can go on and define knowledge operators that involve a group of

agents in some way. This gives rise to the notions of common and (distributed) group knowledge.

The simplest notion is that of 'everybody knows', often denoted by the operator $\mathbf{E_K}$. But one can also add an operator $\mathbf{C_K}$ for 'common knowledge, which is much more powerful. The language is the same as epistemic logic, only now extended with the clause:

DEFINITION 22. (multi-agent epistemic logic.)

- if φ is a multi-agent epistemic formula, then $\mathbf{E_K}\varphi$ and $\mathbf{C_K}\varphi$ are multi-agent epistemic formulas.

For the interpretation we use the following models:

DEFINITION 23.
Models for n-agent epistemic logic are Kripke structures of the form

$$\mathfrak{M} = \langle W, V, R_1, \ldots, R_n, R_E, R_C \rangle$$

where:

- W is a non-empty set of states (or worlds);
- V is a truth assignment function per state;
- The R_i are accessibility relations on W for interpreting the modal operators $\mathbf{K}_i$, assumed to be equivalence relations;
- $R_E = \bigcup_i R_i$;
- $R_C = R_E^*$, the reflexive transitive closure of R_E.

DEFINITION 24. (Interpretation of multi-agent epistemic formulas)
In order to determine whether an multi-agent epistemic formula is true in a model/state pair $\mathfrak{M}, w$ ($\mathfrak{M}, w \models \varphi$), we stipulate:

- $\mathfrak{M}, w \models p$ iff $V(w)(p) = true$, for $p \in \mathcal{P}$;
- the logical connectives are interpreted as usual;
- $\mathfrak{M}, w \models \mathbf{K}_i\varphi$ iff $\mathfrak{M}, w' \models \varphi$ for all w' with $R_i(w, w')$;
- $\mathfrak{M}, w \models \mathbf{E_K}\varphi$ iff $\mathfrak{M}, w' \models \varphi$ for all w' with $R_E(w, w')$;
- $\mathfrak{M}, w \models \mathbf{C_K}\varphi$ iff $\mathfrak{M}, w' \models \varphi$ for all w' with $R_C(w, w')$.

Using the analogous notion of validity as for single-agent epistemic logic, we obtain:

PROPOSITION 25.

- $\models \mathbf{E_K}\varphi \leftrightarrow \mathbf{K}_1\varphi \wedge \ldots \wedge \mathbf{K}_n\varphi$
- $\models \mathbf{C_K}(\varphi \to \psi) \to (\mathbf{C_K}\varphi \to \mathbf{C_K}\psi)$

- $\models \mathbf{C_K}\varphi \to \varphi$
- $\models \mathbf{C_K}\varphi \to \mathbf{C_K}\mathbf{C_K}\varphi$
- $\models \neg\mathbf{C_K}\varphi \to \mathbf{C_K}\neg\mathbf{C_K}\varphi$
- $\models \mathbf{C_K}\varphi \to \mathbf{E_K}\mathbf{C_K}\varphi$
- $\models \mathbf{C_K}(\varphi \to \mathbf{E_K}\varphi) \to (\varphi \to \mathbf{C_K}\varphi)$

The first statement of this proposition shows that the ‘everybody knows’ modality is indeed what its name suggests. The next four says that common knowledge has at least the properties of knowledge: closed under implication, it is true, and enjoys the introspective properties. The sixth property says that common knowledge is known by everybody. The last is a kind of induction principle: the premise gives the condition under which one can ‘upgrade the truth of φ to common knowledge of φ; this premise expresses that it is common knowledge that the truth of φ is known by everybody.

As to multi-agent doxastic logic one can look at similar notions of ‘everybody believes’ and common belief . One can introduce operators $\mathbf{E_B}$ and $\mathbf{C_B}$ for these notions:

DEFINITION 26. (multi-agent doxastic logic)

- if φ is a multi-agent doxastic formula, then $\mathbf{E_B}\varphi$ and $\mathbf{C_B}\varphi$ are multi-agent doxastic formulas

For the interpretation we use the following models:

DEFINITION 27. Models for n-agent doxastic logic are Kripke structures of the form

$$\mathfrak{M} = \langle W, V, R_1, \ldots, R_n, R_F, R_D \rangle$$

where:

- W is a non-empty set of states (or worlds);
- V is a truth assignment function per state;
- The R_i are accessibility relations on W for interpreting the modal operators $\mathbf{B}_i$, assumed to be serial, transitive and euclidean relations;
- $R_F = \bigcup_i R_i$;
- $R_D = R_F^+$, the (nonreflexive) transitive closure of R_F.

Note that the accessibility relation for common belief is the *non*reflexive closure of R_F, contrary to that for common knowledge. This has to do with the fact that common belief needs not to be true!

DEFINITION 28. (Interpretation of multi-agent doxastic formulas.)
In order to determine whether an multi-agent epistemic formula is true in a model/state pair $\mathfrak{M}, w$ ($\mathfrak{M}, w \models \varphi$), we stipulate:

- ... (as usual)
- $\mathfrak{M}, w \models \mathbf{B}_i\varphi$ iff $\mathfrak{M}, w' \models \varphi$ for all w' with $R_i(w, w')$;
- $\mathfrak{M}, w \models \mathbf{E_B}\varphi$ iff $\mathfrak{M}, w' \models \varphi$ for all w' with $R_F(w, w')$;
- $\mathfrak{M}, w \models \mathbf{C_B}\varphi$ iff $\mathfrak{M}, w' \models \varphi$ for all w' with $R_D(w, w')$.

Now we obtain a similar set of properties for common belief (cf. [50]):

PROPOSITION 29.

- $\models \mathbf{E_B}\varphi \leftrightarrow \mathbf{B}_1\varphi \wedge \ldots \wedge \mathbf{B}_n\varphi$
- $\models \mathbf{C_B}(\varphi \to \psi) \to (\mathbf{C_B}\varphi \to \mathbf{C_B}\psi)$
- $\models \mathbf{C_B}\varphi \to \mathbf{E_B}\varphi$
- $\models \mathbf{C_B}\varphi \to \mathbf{C_B}\mathbf{C_B}\varphi$
- $\models \neg\mathbf{C_B}\varphi \to \mathbf{C_B}\neg\mathbf{C_B}\varphi$
- $\models \mathbf{C_B}\varphi \to \mathbf{E_B}\mathbf{C_B}\varphi$
- $\models \mathbf{C_B}(\varphi \to \mathbf{E_B}\varphi) \to (\mathbf{E_B}\varphi \to \mathbf{C_B}\varphi)$

Note the differences due to the fact that common belief is not based on a reflexive accessibility relation.

7.2 Multi-agent BDI logic

Also with respect to the other modalities one may consider multi-agent aspects. In this subsection we focus on the notion of collective or joint intentions. We follow ideas from [29] (but we give a slightly different but equivalent presentation of definitions). We now assume that we have belief and intention operators $\mathbf{B}_i, \mathbf{I}_i$ for every agent $1 \leq i \leq n$. First we enrich the language of multi-agent doxastic logic with operators $\mathbf{E_I}$ (everybody intends) and $\mathbf{M_I}$ (mutual intention). (We call this a multi-agent BDI logic, although multi-agent BI logic would be a more adequate name, since we leave out the modality of desire / goal.)

DEFINITION 30. (multi-agent BDI logic.)
Multi-agent BDI logic is obtained by taking the (analogous clauses of) multi-agent doxastic logic of the previous subsection extended with the clauses:

- if φ is a multi-agent BDI formula, then so is $\mathbf{I}_i\varphi$ for every $1 \leq i \leq n$).
- if φ is a multi-agent BDI formula, then $\mathbf{E_I}\varphi$ and $\mathbf{M_I}\varphi$ are multi-agent BDI formulas.

The language thus obtained is interpreted on slightly enhanced models.

DEFINITION 31.
Models for n-agent BDI logic are Kripke structures of the form

$$\mathfrak{M} = \langle W, V, R_1, \ldots, R_n, R_F, R_D, S_1, \ldots, S_n, S_F, S_D \rangle$$

where:

- W is a non-empty set of states (or worlds);
- V is a truth assignment function per state;
- The R_i are accessibility relations on W for interpreting the modal operators $\mathbf{B}_i$, assumed to be serial, transitive and euclidean relations, while the S_i are accessibility relations on W for interpreting the modal operators $\mathbf{I}_i$, assumed to be serial relations.
- $R_F = \bigcup_i R_i$ and $S_F = \bigcup_i S_i$;
- $R_D = R_F^+$ and $S_D = S_F^+$, the (nonreflexive) transitive closure of R_F and S_F, respectively.

DEFINITION 32. (Interpretation of multi-agent BDI formulas.) In order to determine whether an multi-agent epistemic formula is true in a model/state pair $\mathfrak{M}, w$ ($\mathfrak{M}, w \models \varphi$), we stipulate:

- $\mathfrak{M}, w \models \mathbf{I}_i\varphi$ iff $\mathfrak{M}, w' \models \varphi$ for all w' with $S_i(w, w')$;
- $\mathfrak{M}, w \models \mathbf{E_I}\varphi$ iff $\mathfrak{M}, w' \models \varphi$ for all w' with $S_F(w, w')$;
- $\mathfrak{M}, w \models \mathbf{M_I}\varphi$ iff $\mathfrak{M}, w' \models \varphi$ for all w' with $S_D(w, w')$;

Hence we get similar properties for mutual intention as we had for common belief (but of course no introspective properties):

PROPOSITION 33.

- $\models \mathbf{E_I}\varphi \leftrightarrow \mathbf{I}_1\varphi \wedge \ldots \wedge \mathbf{I}_n\varphi$;
- $\models \mathbf{M_I}(\varphi \to \psi) \to (\mathbf{M_I}\varphi \to \mathbf{M_I}\psi)$;
- $\models \mathbf{M_I}\varphi \to \mathbf{E_I}\varphi$;
- $\models \mathbf{M_I}\varphi \to \mathbf{E_I}\mathbf{M_I}\varphi$;
- $\models \mathbf{M_I}(\varphi \to \mathbf{E_I}\varphi) \to (\mathbf{E_I}\varphi \to \mathbf{M_I}\varphi)$.

We see that E-intentions ('everybody intends') and mutual intentions are defined in a way completely analogous with E-beliefs ('everybody believes') and common beliefs, respectively.

Next we define the notion of *collective intention* ($\mathbf{C_I}$) as follows:

DEFINITION 34.

- $\mathbf{C_I}\varphi \;=\; \mathbf{M_I}\varphi \wedge \mathbf{C_B}\mathbf{M_I}\varphi$

This definition states that collective intentions are those formulas that are mutually intended and of which this mutual intention is a common belief amongst all agents in the system.

Furthermore, we mention here that in the literature there is also other work on BDI-like logics for multi-agent systems where we encounter such notions as joint intentions, joint goals and joint commitments, mostly coined in the setting of how to specify teamwork. Seminal work was done by Cohen & Levesque [19]. This work was a major influence on our own multi-agent version of KARO [2]. An important complication in a notion of joint goal involves that of persistence of the goal: where in the single agent case the agent pursues its goal until it believes it has achieved it or believes it can never be achieved, in the context of multiple agents, the agent that realizes this, has to inform the others of the team about it so that the group / team as a whole will believe that this is the case and may drop the goal. Also the work of Singh [71] must be mentioned here, where an interesting distinction is made between *exodeictic* and *endodeictic* intentions of groups, where the former is 'pointing outward' (intention of the group as viewed by others) while the latter is 'pointing inward' (intention as viewed by the group itself).

8 FURTHER DEVELOPMENTS

One may wonder where the development of agent logics is heading. Moreover, aren't these logics which are ever increasing in expressive power, becoming much too complex in terms of computational complexity to be used in practice? To begin with the latter: yes, one may indeed expect higher complexity when stacking modalities (cf. [44]), but whether this is considered a bad thing, depends on the application. It is my impression that logics for (particularly single) agents have been employed to try and coin philosophical concepts more concretely and give them a precise meaning within the logical formalisms that have been proposed. They were not meant for practical reasoning about agents in the first instance. After having done this work, agent researchers such as Rao & Georgeff turned to devising practical implementations of agent systems, employing the concepts they studied in the logics in a much more practical way (e.g. databases of beliefs and goals instead of modal operators), which begs the very important and not yet completely resolved question how the formal and practical notions of e.g. BDI relate to each other. Other people tried to render their logical specifications directly executable by considering fragments of modal/temporal logics (e.g. the researchers working on the METATEM language [33]). On the other hand there is also some positive (or at least not too negative) news even for those who are interested in the computational complexity of the theoretical BDI-like logics. In [64] it is reported that the complexity of tableau-based decision procedures for propositional logic is not higher than the complexity of the underlying temporal logic (which is, admittedly, quite hard, viz. exponential in the size of the input formula). Furthermore, one may also consider restricting the languages of combined modal formalisms to reduce complexity, but, of course, this will generally come with a reduction of expressive power as well, and it depends on the applications whether this is sensible. We refer to [5] for a discussion of both (positive and negative) results on the complexity of combined modal logics and possible ways to respond to these results. As said before, concerning logics for specifying the behavior and attitudes of single agents one can observe that recent work has turned away from purely logical theories towards more implementation-oriented approaches. For example, research on agent-oriented programming languages such as Agent0 [70], AgentSpeak(L) [63], and 3APL [45], and agent-oriented software engineering (AOSE, [94]), more generally. Of course, as in software engineering in general, there is a need to verify that the software

one has developed is correct in the sense that it behaves as specified *a priori*. In the field of agent-oriented programming this has sparked off research on correctness logics that are adequate in this context. Naturally, these logics are very close to the logics we have seen in this chapter: if one programs in terms of cognitive notions such as beliefs, desires, goals, plans, etc., it is obvious that one also needs to reason about these notions when one tries to verify such a program! For example, Bordini *et al.* [7] work on the use of model-checking techniques for a linear-time temporal logic in which properties of programs written in AgentSpeak can be specified. We ourselves have initiated work on the correctness of (fragments of) 3APL [82]. Time will tell how successful these approaches will be. As for logics for multi-agent systems: if one considers 'societies of agents' obviously also other notions become important besides mere multi-agent extensions of -notions. For instance, one can investigate how communication takes place in such a system, and how this affects the mental states of the agents in the system. This, in turn, is important for synchronization, coordination, and cooperation in the system. There has also been done some work on this. As mentioned before, Dignum and Van Linder [28] have extended the KARO framework to deal with speech acts. Moreover, in societies it may be important to consider norms, obligations and permissions as a way to control societal behavior.

In a series of papers [12, 11, 10] Van der Torre *et al.* have extended the BDI framework to what they call the BOID framework dealing with the Beliefs, Obligations, Intentions and Desires of agents. Although the language of BOID contains operators for belief (B), obligation (O), intention (I) and desire (D), and thus looks like an amalgam of BDI and deontic logic, BOID logic is not really a modal logic in the proper sense. The operators are not interpreted by means of accessibility relations in Kripke models. Instead, a default logic [65] is employed and a BOID agent is specified by a number of default rules involving the BOID notions/operators, together with a priority relation on these rules. The form of these rules is $X_1 \hookrightarrow X_2$, where X_1, X_2 typically are expressions of the form $\mathrm{B}\varphi$, $\mathrm{O}\varphi$, $\mathrm{I}\varphi$, or $\mathrm{D}\varphi$. The main concern is which (consistent) extensions are yielded representing how the beliefs, obligations, intentions and desires can be combined (consistently) taking the priority on rules into account. Another extension with a similar philosophy in mind of incorporating social notions into the BDI framework was proposed by Dignum *et al.* [27, 22]. This framework is called B-DOING, and treats Beliefs, Desires, Obligations, Intentions, Norms and Goals. This approach is more like a normal modal logic, although a number of extra elements is added. As to the deontic aspect, this framework is built on dyadic (conditional) obligations. Logically, the most important addition is the incorporation of two operators: $N^z(p|q)$ and $O^x_{ab}(p|q)$, with as intended meanings 'it is a norm of the society / organization z that p should be true when q is true' and 'when q is true, individual a is obliged to b that p should be true, where z is the organization/society that is responsible for enforcing the penalty', respectively. Formally, to give an interpretation to these operators a possible world semantics is employed that is rather involved. The upshot of this semantics is that $N^z(p|q)$ ($O^z(p|q)$) holds if $p \wedge q$ worlds are preferred to $\neg p \wedge q$ ones and the (maximally) preferred q worlds satisfy p, where the preference relation on worlds is induced by (associated with) the norms and obligations in organization z, respectively. Related to this is a development in the implementation of multi agent systems where one tries to restrain/constrain individual agents by means of an (electronic) institution [30]). The idea here is that agents in a MAS or agent society must obey certain norms and to assist them in doing this certain protocols are devised that the agents in the system are advised or even enforced to follow

in order to abide by the norms. This calls for questions as to the relation between the typically abstract norms and the typically very concrete protocols, and how following the protocols guarantees non-violation of the norms. These questions are partly of a logical nature. For instance, how may a particular step of the protocol *count as* an implementation of an abstract norm [47]? This is of vital importance if one wants to verify whether the protocol guarantees norm compliance. Interestingly, also the logic of 'counts-as' can be put into a modal logic setting [39], opening up a new area of modal logic applications. In conclusion we can state that the application area of modal logic to reason about multi-agent systems is still flourishing, and many interesting problems remain to be investigated and solved...!

9 COUNTERFACTUAL CONDITIONALS

A chapter on modal logics in Artificial Intelligence should contain a section in which conditional logics are discussed, because conditionals logics, in particular the logics developed for counterfactual conditionals, pop up everywhere in AI where non-monotonicity plays a role.

Counterfactual conditionals are sentences of the form

(1) $\ulcorner$*If it had been the case that* φ*, it would have been the case that* $\psi\urcorner$

They are typically uttered in contexts where the antecedent is false and known to be false. Therefore, they cannot be analyzed as material implications, because material implications with a false antecedent are true no matter what the consequent says.

Counterfactuals cannot be analyzed as strict implications either. One cannot equate a sentence of the form given in 1 with a formula of the form

(2) $\Box(\varphi \to \psi)$

where $\Box$ is the necessity operator of any normal system of modal logic, because any such system validates logical principles that do not hold for counterfactuals. One such principle is *Strengthening the Antecedent.* In any extension of **K**, we have

$$\Box(\varphi \to \chi) \models \Box((\varphi \wedge \psi) \to \chi)$$

However, from

(3) *If I had put sugar in my coffee, it would have tasted better,*

it does not follow that

(4) *If I had put sugar and diesel oil in my coffee, it would have tasted better.*

The starting point for the discussion in the following sections is the analysis of counterfactuals developed by Robert Stalnaker [72] and David Lewis [52]. Roughly put, they proposed the following truth condition for counterfactual conditionals.

- A sentence of the form $\ulcorner$*If it had been the case that* φ*, it would have been the case that* $\psi\urcorner$ is true in the actual world w iff the consequent ψ is true in all accessible worlds in which (a) the antecedent φ is true, and which (b) in other respects differ minimally from w.

In other words, the consequent ψ need not be true in *all* accessible worlds in which the antecedent φ is true, which it would have to be if counterfactuals were strict implications. What matters is ψ's truth value in a particular subset of this set, the φ-worlds that are most similar to the actual world. It is easy to see how this semantics blocks the inference from 3 to 4. Consider the set S of worlds in which (i) *I put sugar in my coffee* is true and which (ii) in other respects differ minimally from the actual world. Presumably, *I put diesel oil in my coffee* is false in all these worlds. Given this, the set T of worlds in which (i) *I put sugar and diesel oil in my coffee* is true, but which (ii) in other respects differ minimally from the actual world will not be a subset of S. Now, *the coffee tastes fine* could very well be true in every world in S, but false in some of the worlds in T.

Let us get more precise. In the sequel we are interested in languages, frames and models that are built up as follows.

- Extend the languages of propositional logic with a new binary operator $\rightsquigarrow$. Until further notice we will read '$\varphi \rightsquigarrow \psi$' as '*If it had been the case that φ, it would have been the case that ψ*'.

- Interpret the resulting languages in frames $\mathfrak{F} = \langle W, \prec \rangle$, where (i) $W \neq \emptyset$ and (ii) $\prec$ is a function which assigns to every $w \in W$ a strict partial ordering $\prec_w$ on some subset W_w of W. The elements of W will play the role of possible worlds. Until further notice the strict partial ordering $\prec_w$ is meant to play the role of a comparative similarity relation; read '$u \prec_w v$' as ' u is more similar to w than v'. The field W_w of this relation $\prec_w$ is the set of worlds that are accessible from w. Inaccessible worlds, i.e. the worlds outside W_w are supposed to be so unlike w that in w it is absurd to assume that the real world might have been be one of those.

- Supply a frame with a valuation V which assigns a truth value to every atomic sentence in every world to get a model $\mathfrak{M} = \langle W, \prec, V \rangle$. As elsewhere in this book, '$\mathfrak{M}, w \models \varphi$' is used to indicate that the formula φ is true in the world w (of the model $\mathfrak{M}$). I will write '$[\![\varphi]\!]_{\mathfrak{M}}$' to refer to $\{w \in W \mid \mathfrak{M}, w \models \varphi\}$, and call this set the proposition expressed by φ (in $\mathfrak{M}$). When it is clear which model $\mathfrak{M}$ is at stake the subscript '$\mathfrak{M}$' in $[\![\varphi]\!]_{\mathfrak{M}}$ will be omitted. Worlds in $[\![\varphi]\!]_{\mathfrak{M}}$ will be called $[\![\varphi]\!]$-worlds.

- Add the following clause to the list of truth conditions for the standard connectives.

  $\mathfrak{M}, w \models \varphi \rightsquigarrow \psi$ iff for every $u \in W_w \cap [\![\varphi]\!]$ the following holds:
  there is some $u' \in [\![\varphi]\!]$ such that $u' \preceq_w u$ and $\mathfrak{M}, u'' \models \psi$ for every $u'' \in [\![\varphi]\!]$ such that $u'' \preceq_w u'$.

Part of the complexity of this truth condition is due to the fact that the partial orders introduced above do not have to satisfy the so-called

Limit Assumption : For every $w \in W$, the relation $\prec_w$ is well-founded.

Call any $u \in U$ a *closest* U-world to w iff $u \in W_w \cap U$ and there is no $v \in U$ such that $v \prec_w u$. Given the Limit Assumption we can be sure that in every non empty subset U of W_w we can find some worlds that are closest to w. This enables us to reformulate the truth condition in a more perspicuous way.

- Suppose the frame $\mathfrak{F} = \langle W, \prec \rangle$ satisfies the Limit Assumption, and consider the model $\mathfrak{M} = \langle W, \prec, V \rangle$. The following holds:

  $\mathfrak{M}, w \models (\varphi \rightsquigarrow \psi)$ iff $\mathfrak{M}, u \models \psi$ for every closest $[\![\varphi]\!]$-world u to w.

Is it reasonable to assume that the comparative similarity relation is well-founded? Are there propositions $[\![\varphi]\!]$ such that for every $[\![\varphi]\!]$-world u some $[\![\varphi]\!]$world v exists that is more similar to w than u is — so that one can get closer and closer to w without ever getting in a $[\![\varphi]\!]$-world that is closest to w? It is not difficult to think of examples. How tall would you be in the closest world in which you are taller than you actually are?

The logic generated by the semantics sketched above is given by the following axioms and rules:

$$\begin{array}{rl}
\text{(CI)}: & \vdash \varphi \rightsquigarrow \varphi \\
\text{(CC)}: & \vdash ((\varphi \rightsquigarrow \psi) \wedge (\varphi \rightsquigarrow \chi)) \rightarrow (\varphi \rightsquigarrow (\psi \wedge \chi)) \\
\text{(CW)}: & \vdash (\varphi \rightsquigarrow \psi) \rightarrow (\varphi \rightsquigarrow (\psi \vee \chi)) \\
\text{(ASC)}: & \vdash ((\varphi \rightsquigarrow \psi) \wedge (\varphi \rightsquigarrow \chi)) \rightarrow ((\varphi \wedge \psi) \rightsquigarrow \chi) \\
\text{(AD)}: & \vdash ((\varphi \rightsquigarrow \chi) \wedge (\psi \rightsquigarrow \chi)) \rightarrow ((\varphi \vee \psi) \rightsquigarrow \chi) \\
\text{(MP}\rightarrow\text{)}: & \varphi \rightarrow \psi, \varphi \vdash \psi \\
\text{(REA)}: & \text{If } \vdash \varphi \leftrightarrow \psi, \text{ then } \varphi \rightsquigarrow \chi \vdash \psi \rightsquigarrow \chi
\end{array}$$

Here, (CI) is short for *Conditional Identity* , (CC) for *Conjunction of Consequents*, (CW) for *Weakening the Consequent*, (SAC) for *Strengthening the Antecedent with a Consequent*, (AD) for *Disjunction of Antecedents*, (MP$\rightarrow$)for *Modus Ponens for* $\rightarrow$, and (REA) for *Replacement of Equivalent Antecedents.*[6]

This system, called **P**, is for conditional logic what **K** is for modal logic: it is the minimal system, which you get if you assume that the relations $\prec_w$ are just partial orderings[7] and have no additional properties. That **P** is (weakly) complete with respect to the class of partial orders was first proved by Burgess in [14]. This proof has been simplified by Friedman and Halpern in [41]. An altogether different proof of (strong) completeness is given in Veltman[84].

If in **P** the scheme (ASC) is strengthened to

$$\textit{Strengthening the Antecedent (AS)}: \quad (\varphi \rightsquigarrow \chi) \rightarrow ((\varphi \wedge \psi) \rightsquigarrow \chi)$$

one gets the system $\mathbf{K}^{\rightsquigarrow}$, which is just **K** in disguise.

PROPOSITION 35.

1. *In the language of modal logic, define* $\varphi \rightsquigarrow \psi$ *by* $\Box(\varphi \rightarrow \psi)$. *Suppose* $\Delta \cup \{\varphi\}$ *consists of formulas of the language of conditional logic. Then* $\Delta \vdash_{\mathbf{K}^{\rightsquigarrow}} \varphi$ *iff* $\Delta \vdash_{\mathbf{K}} \varphi$.

2. *In the language of conditional logic, define* $\Box\varphi$ *as* $\neg\varphi \rightsquigarrow \bot$. *Suppose* $\Delta \cup \{\varphi\}$ *consist of formulas of the language of modal logic.* $\Delta \vdash_{\mathbf{K}} \varphi$ *iff* $\Delta \vdash_{\mathbf{K}^{\rightsquigarrow}} \varphi$.

It is straightforward to prove (i) and (ii) from left to right. To prove (i) from right to left use (ii) from left to right, and similarly for (ii) from right to left use (i) from left to right.

[6] Given CC and WW, one does not need a separate Replacement rule for Equivalent Consequents to prove the full Replacement Rule: If $\vdash \varphi \leftrightarrow \psi$, then $\chi \vdash \chi'$ where χ' is the result of substituting φ for ψ at one or more places where ψ occurs in χ.

[7] Actually, the only property that matters is transitivity. Irreflexivity is not expressible.

Does the Limit Assumption make a difference to the logical properties of $\rightsquigarrow$? It does, but only for arguments with infinitely many premises. Under the Limit Assumption compactness fails:

PROPOSITION 36.
Let $p_1, \ldots, p_n, \ldots$ be countably many distinct atomic sentences, and let φ_k, and ψ_k, for any k be defined as follows:

$$\varphi_k = ((p_1 \vee \ldots, p_{k+1}) \rightsquigarrow \neg(p_1 \vee \ldots, p_k))$$

$$\psi_k = \neg((p_1 \vee \ldots, p_{k+1}) \rightsquigarrow (p_1 \vee \ldots, p_k))$$

Consider the set Δ consisting of all φ_k's and ψ_k's. The Limit Assumption holds iff Δ is not satisfiable.

So far no constraints have been imposed on the comparative similarity relation $\prec$ that distinguish it from any other other relation that holds between three objects u, v and w when 'u is more ... to w than v. What extras does the fact one has to fill the dots with the word 'similar' bring?

Weak Centering: $w \in W_w$ for every $w \in W$, and for no $v \in W_w$ it holds that $v \prec_w w$.

Imposing this constraint means the next rule gets valid.

$$\textit{Modus Ponens for} \rightsquigarrow (\mathrm{MP}^{\rightsquigarrow}): \quad \varphi \rightsquigarrow \psi, \varphi \vdash \psi$$

Weak centering says that no world can be closer to a world w than w itself. If in addition you think that no world different from w can be equally close to w as w itself, you get this.

Strong Centering: $w \in W_w$ for every $w \in W$, and for every $v \in W_w$ such that $v \neq w$, $w \prec_w v$.

The logical pay off is this:

$$\textit{Conjunctive Sufficiency}: \quad (\varphi \wedge \psi) \rightarrow (\varphi \rightsquigarrow \psi)$$

If in establishing similarities and dissimilarities *all* characteristics of the worlds are taken into consideration, one of the consequences will be that only the world w itself will resemble the world w as much as the world w does. But in cases in which only *some* characteristics matter, there will often be more than one world that is just like w in all relevant respects. In these cases the structures will satisfy *Weak Centering*, but not *Strong Centering*

If you believe that two different worlds cannot be equally close to the actual one, you will support the following constraint:

$$\textit{Connectedness:} \quad \text{for any } u, v \in W_w, \text{ either, } u = w, \text{ or } u \prec_w v, \text{ or } v \prec_w u.$$

In the presence of the Limit Assumption Connectedness implies that there will always be for any antecedent φ at most one $[\![\varphi]\!]$-world most resembling the actual world. This uniqueness assumption brings the following principle in its train:

$$\textit{Conditional Excluded Middle } (\mathrm{CEM}): \quad (\varphi \rightsquigarrow \psi) \vee (\varphi \rightsquigarrow \neg\psi)$$

Couldn't there be cases where we have several $[\![\varphi]\!]$-worlds, all equally close to the actual world and all closer to the actual world than any other world? In [52] Lewis brings in the following example, due to W.V.O. Quine, to show that such cases do exist:

(5) *If Bizet and Verdi had been compatriots, Bizet would have been Italian.*

(6) *If Bizet and Verdi had been compatriots, Verdi would have been French.*

Now, if there is only one world closest to the actual world in which Bizet and Verdi are compatriots, it is impossible that both (5) and (6) are false while (7) is true:

(7) *If Bizet and Verdi had been compatriots, either Verdi would have been French or Bizet would have been Italian.*

According to Lewis one can accept (7) without having to accept (5) or (6), and so he rejects the uniqueness assumption.

Lewis does accept the following constraint:

Almost-Connectedness: for any $u, v, w \in W_z$, if $u \prec_z w$, then either $u \prec_z v$ or $v \prec_z w$.

Define $u \simeq_w v$ iff neither $u \prec_w v$ nor $u \prec_w v$. The relation $\simeq_w$ is reflexive, and symmetric, but not necessarily transitive. Requiring that the relation $\prec_w$ is almost connected amounts to requiring that $\simeq_w$ is transitive. In that case we can read '$u \simeq_w v$' as 'u and v are equally similar to w', and we can picture the relation $\prec_w$ as a linear order of equivalence classes of worlds. The corresponding axiom scheme is this:

Strengthening with a Possibility (ASP): $(\neg(\varphi \rightsquigarrow \neg\psi) \wedge (\varphi \rightsquigarrow \chi)) \rightarrow ((\varphi \wedge \psi) \rightsquigarrow \chi)$

The axiom ASP says that an antecedent of a counterfactual $\varphi \rightsquigarrow \chi$ may be strengthened with a formula ψ provided that the counterfactual assumption φ does not exclude the possibility that ψ. So, given the validity of ASC, this leaves only one case in which it is not allowed to strengthen the antecedent of a counterfactual $\varphi \rightsquigarrow \chi$ with the formula ψ. That's when $\varphi \rightsquigarrow \neg\psi$ is true and $\varphi \rightsquigarrow \psi$ is false. In the other three cases:

1. $\varphi \rightsquigarrow \psi$ is true, $\varphi \rightsquigarrow \neg\psi$ is true
2. $\varphi \rightsquigarrow \psi$ is true, $\varphi \rightsquigarrow \neg\psi$ is false
3. $\varphi \rightsquigarrow \psi$ is false, $\varphi \rightsquigarrow \neg\psi$ is false

strengthening the antecedent φ with ψ is valid.

Is it reasonable to assume that the comparative similarity relation is almost connected? Everybody who has tried to analyze the notion of comparative similarity and to explain how it comes about, concluded that it is not.[8] Still, it is not easy to find a convincing counterexample to ASP. Ginsberg [38] suggests:

It's not the case that if Verdi and Satie had been compatriots, Satie and Bizet would not have been compatriots.
If Verdi and Satie had been compatriots, Bizet would have been French

---

If both Verdi and Satie, and Satie and Bizet had been compatriots, Bizet would have been French.

Despite this counterexample and the theoretical arguments underlying it, presently the most popular system for counterfactuals is given by $\mathbf{P}$ + ASP + $\mathrm{MP}^{\rightsquigarrow}$.

---

[8]For a critical analysis of the notion comparative similarity see Fine[32], Veltman[83], [86], Tichy[73], Pollock[61], Lewis[53], Kratzer[48], Kratzer[49].

10 NON-MONOTONIC CONSEQUENCE RELATIONS

The standard model theoretic notion of logical validity is monotonic: if ψ follows from $\varphi_1, \ldots, \varphi_n$, then ψ follows from $\varphi_1, \ldots, \varphi_n, \varphi_{n+1}$. This is so, because the standard notion requires that the conclusion be true in *any* model in which the premises are true, and, clearly, if ψ is true in *any* model in which $\varphi_1, \ldots, \varphi_n$ are true, then certainly so in *any* model in which $\varphi_1, \ldots, \varphi_n$ plus φ_{n+1} are true.

Non-monotonic logic started when in the late seventies logicians working in Artificial Intelligence noticed that in many practical situations when people draw a conclusion, they do not reckon with all conceivable possibilities left open by the premises, but only with some of these, the *most normal* ones or the ones *most likely* to occur. Something similar happened in the field of epistemic logic when at some point one got interested in arguments in which the premises represent 'all that is known'. In such cases the question is not so much whether the conclusion holds in all situations in which the premises hold, but whether it holds in the *'most ignorant'* situations among these.

There are more examples in which the phrase *'any model'* occurring in the definition of the standard notion of validity is restricted to 'the most ... models', where the dots are to be filled by some adjective. All these alternative notions of validity can be formally captured by assuming that the models of the language are ordered by a well-founded partial ordering $\prec$ and to stipulate that ψ is a (non-monotonic) consequence of $\varphi_1, \ldots, \varphi_n$ iff ψ is true in all models that are $\prec$-minimal in the class of models in which the premises $\varphi_1, \ldots, \varphi_n$ are true.

This must remind the reader of the frames and the truth-condition for counterfactuals introduced in the preceding section. Indeed, we are dealing here with a special case of the framework introduced there. In addition to the Limit Assumption, the following constraints are at stake.

Universality:	for every $w \in W, W_w = W$.
Absoluteness:	for every $u, w \in W, \prec_u = \prec_w$.

Absoluteness says that the relation $\prec_w$ is in fact independent of w, so that one can omit the subscript. *Universality* adds that $\prec$ is an ordering of the set of all possible worlds. So, the relations $\prec_w$ are all equal to one and the same well-founded partial ordering $\prec$ of the set of all possible worlds.

Secondly, given *Universality* and *Absoluteness*, if a sentence of the form $\varphi \rightsquigarrow \psi$ is true in one world of a model $\mathfrak{M}$, it will in fact be true in every world of $\mathfrak{M}$. This means that the following holds:

$$\mathfrak{M} \models \varphi \rightsquigarrow \psi \text{ iff } \mathfrak{M}, w \models \psi \text{ for every } \prec\text{-minimal world } w \text{ in } [\![\varphi]\!].$$

Finally, let's write '$\varphi_1, \ldots, \varphi_n \mid\!\sim \psi$' instead of '$(\varphi_1 \wedge \ldots \wedge \varphi_n) \rightsquigarrow \psi$', and '$\varphi_1, \ldots, \varphi_n \mid\!\sim_{\mathfrak{M}} \psi$' instead of '$\mathfrak{M} \models (\varphi_1 \wedge \ldots \wedge \varphi_n) \rightsquigarrow \chi$'. In doing so, we arrive at what in Kraus et al.[51] appears as the definition of 'the entailment relation $\mid\!\sim_{\mathfrak{M}}$ defined by the model $\mathfrak{M}$'.

(*) $\qquad \varphi_1, \ldots, \varphi_n \mid\!\sim_{\mathcal{M}} \psi$ iff $\mathfrak{M}, w \models \psi$ for every $\prec$-minimal world w in $[\![\varphi_1]\!] \cap \ldots \cap [\![\varphi_n]\!]$.

The authors of [51] refer to the relation $\prec$ as a preference relation, and to the models $\mathfrak{M} = \langle W, \prec, V \rangle$ as preferential models. They are interested in the properties of the *preferential* consequence relation $\mid\!\sim$, formally modeled by (*).

It will come as no surprise that $\mid\!\sim$ behaves like a counterfactual implication $\rightsquigarrow$. However, there is an important syntactic difference between between $\mid\!\sim$ and $\rightsquigarrow$. Conditionals sometimes occur nested in other conditionals — as in $\varphi \rightsquigarrow (\psi \rightsquigarrow \chi)$ — but nesting sentences expressing an entailment relation is quite incomprehensible. The entailment relation belongs to the metalanguage rather than the object language. What could $\varphi \mid\!\sim (\psi \mid\!\sim \chi)$ possibly mean?

This, however, does not give rise to important semantic differences between $\mid\!\sim$ and $\rightsquigarrow$.

PROPOSITION 37.
Let Δ be a set of formulas containing only non-nested conditionals. If Δ is satisfiable on any frame, then it is satisfiable on a frame with a universal and absolute $\prec$ relation.[9]

Given this, one might expect the system **P** to give a complete characterization of the properties of $\mid\!\sim$. Kraus et al.[51], using the methods of [84], prove that this is indeed the case. One easily recognizes the axiom schemes introduced in the previous section in the next principles of entailment.

$$\begin{array}{rl}
\text{(CI) becomes } \textit{Reflexivity}: & \varphi \mid\!\sim \varphi \\
\text{(CC) becomes } \textit{And}: & \text{If } \varphi \mid\!\sim \psi \text{ and } \varphi \mid\!\sim \chi, \text{ then } \varphi \mid\!\sim (\psi \wedge \chi) \\
\text{(CW) becomes } \textit{Right Weakening}^{10}: & \text{If } \varphi \mid\!\sim \psi \text{ and } \psi \models \chi, \text{ then } \varphi \mid\!\sim \chi \\
\text{(ASC) becomes } \textit{Cautious Monotony}: & \text{If } \varphi \mid\!\sim \psi \text{ and } \varphi \mid\!\sim \chi, \text{ then } \varphi, \psi \mid\!\sim \chi \\
\text{(AD) becomes } \textit{Or}: & \text{If } \varphi \mid\!\sim \chi \text{ and } \psi \mid\!\sim \chi, \text{ then } (\varphi \vee \psi) \mid\!\sim \chi \\
\text{(REA) becomes } \textit{Left Logical Equivalence}: & \text{If } \varphi \models \psi \text{ and } \psi \models \varphi, \text{ then} \\
& \quad \text{if } \varphi \mid\!\sim \chi, \text{ also } \psi \mid\!\sim \chi
\end{array}$$

The literal translation of (CW) would be ' If $\varphi \mid\!\sim \psi$, then $\varphi \mid\!\sim \psi \vee \chi$'. *Right Weakening* is equivalent to this. ($\models$ stands for the classical entailment relation.)

As a characterization of an entailment relation the system **P** is a bit odd. One would expect only purely structural principles. The principles *Or*, and *And*, however, presuppose that the object language has connectives with the properties of conjunction and disjunction. Kraus et al.[51] also discuss a weaker system consisting of only structural rules. It is called **C**, where '**C**' stands for 'cumulative', and it was originally proposed by Dov Gabbay[34] as a system describing the weakest reasonable consequence relation. It is given by: *Reflexivity, Right Weakening, Cautious Monotony, Left Logical Equivalence*, and

$$\textit{Cut}: \quad \text{If } \varphi, \psi \mid\!\sim \chi \text{ and } \varphi \mid\!\sim \psi, \text{ then } \varphi \mid\!\sim \chi$$

It is left to the reader to show that *Cut* is a derived rule of **P**.

An important field in which a non-monotonic consequence relation is employed is the field of default reasoning. Actually, in the modal approach to default reasoning not only the consequence relation but also the defaults rules themselves are modeled after conditionals. Read '$\varphi \rightsquigarrow \psi$' as '*If φ, then normally ψ*', and take the underlying well-founded ordering $\prec$ of the set of possible worlds to be the relation '*... is more normal than...*'. Then a rule $\varphi \rightsquigarrow \psi$ will hold in a model if ψ is true at the most normal $[\![\varphi]\!]$-worlds. An agent

[9]This proposition does not hold for arbitrary sets of formulas. If nesting is allowed one has to add the **S5** axioms $\Box\varphi \to \varphi$, $\Box\varphi \to \Box\Box\varphi$, and $\Diamond\varphi \to \Box\Diamond\varphi$ to **P** in order to get a system that is complete with respect to the universal and absolute frames. (Here $\Box\varphi =_{df} \neg\varphi \rightsquigarrow \bot$.)

who has learnt that φ is the case and who accepts the rule $\varphi \rightsquigarrow \psi$ will expect that ψ is the case provided there is no evidence that the case at hand is exceptional.

More generally, default rules are of crucial importance when some decision must be made in circumstances where the facts of the matter are only partly known. In such a case one must reckon with several possibilities. Default rules serve to narrow down this range of possibilities: some of these possibilities are more normal than other. An agent will expect that the actual world conforms to as many standards of normality as possible given the information at hand.

Several theories have been developed that formalize this phenomenon. They differ in the way they formally capture the idea that an agent will expect the actual world to be as normal as possible given the circumstances described by the premises. James Delgrande [21] was the first who proposed a definition for the set of worlds that best meet the agent's expectations. Alternative definitions are proposed in Asher & Morreau[4] and Veltman [85]. See [6] for a detailed comparison of these theories and Halpern et.al.[41] for technical insights.

11 BELIEF REVISION

There is still another way to read $\varphi \rightsquigarrow \psi$: 'After a revision by φ, it is believed that ψ'. Here the topic is belief revision, and the question at stake is how an agent should change his or her beliefs in the face of new information. The formula φ is supposed to bring new information — possibly contradicting the information available — and if $\varphi \rightsquigarrow \psi$ is true, this means that ψ is accepted after the incorporation φ in ones stock of beliefs.

Checking the axioms for $\rightsquigarrow$ with this reading in mind, we find that many of them sound quite plausible. For example: *Conditional Identity*, $\varphi \rightsquigarrow \varphi$, becomes *'After a revision by φ, it is believed that φ'*, and *Disjunction of Antecedents*, $(\varphi \rightsquigarrow \chi) \wedge (\psi \rightsquigarrow \chi)) \rightarrow ((\varphi \vee \psi) \rightsquigarrow \chi)$, can be read as '*If both a revision with φ and a revision by ψ lead to the belief χ, then so does a revision by $\varphi \vee \psi$*'. Are we here once more dealing with **P** or one of its extensions?

Let's start at the beginning. In 1985 Carlos Alchourron, Peter Gärdenfors and David Makinson published a by now classic paper [1] in which they discuss three forms of belief change: expansion, contraction and revision. Modeling an agents beliefs by a deductively closed theory K, called a belief set, a number of rationality postulates are laid down for the expansion K^+_φ of K by φ, the contraction K^-_φ of K by φ, and the revision K^*_φ of K by φ.

The constraints for expansion uniquely determine K^+_φ as the set $\{\psi \mid K, \varphi \vdash \psi\}$. The constraints for contraction and revision do not uniquely determine K^-_φ and K^*_φ because the outcomes of these operations do not depend on logical factors only. Epistemic factors may also play a role. For example, in revising their beliefs agents may be prepared to give up one sentence rather than the other because the empirical support for the one is much better than for the other.

Here are the so-called AGM postulates for revision as formulated in Gärdenfors[35]:

$\mathbf{K}^*1$ For any sentence φ and any belief set K, K^*_φ is a belief set

$\mathbf{K}^*2$ $\varphi \in K^*_\varphi$

$\mathbf{K}^*3$ $K^*_\varphi \subseteq K^+_\varphi$

K*4 If $\neg\varphi \notin K$, then $K^+_\varphi \subseteq K^*_\varphi$

K*5 $K^*_\varphi = \{\psi \mid \bot \vdash \psi\}$ iff $\vdash \neg\varphi$;

K*6 If $\vdash \varphi \leftrightarrow \psi$, then $K^*_\varphi = K^*_\psi$

K*7 $K^*_{\varphi\wedge\psi} \subseteq (K^*_\varphi)^+_\psi$

K*8 If $\neg\varphi \notin K$, then $(K^*_\psi)^+_\varphi \subseteq K^*_{\varphi\wedge\psi}$

Adam Grove[40] was the first to notice that the semantics for counterfactuals as defined in section 9, supplies an interpretation for these postulates. For every belief set K, we consider the set of *models for* K, where a model $\mathfrak{M}_K$ for K is given by $\mathfrak{M}_K = \langle W, \prec, V\rangle$, where

- W is the set of all maximal consistent theories of the language in which K is formulated;
- $\prec$ is a well-founded and almost connected strict partial ordering of W such that the $\prec$-minimal elements of W are given by the set of maximal consistent extensions of K;
- $V(p)(w) = 1$ iff $p \in w$.

PROPOSITION 38.
Let $\mathfrak{M}_K = \langle W, \prec, V\rangle$ be a model for K. Define K^*_φ for every φ as follows:

$$\psi \in K^*_\varphi \textit{ iff } \psi \in w \textit{ for every } w \textit{ such that } w \textit{ is } \prec\textit{-minimal in } [\![\varphi]\!].$$

Then the postulates **K*****1** *to* **K*****8** *are satisfied.*

Conversely, we have

PROPOSITION 39.
Let K^* be a revision function for some belief set K satisfying **K*****1** *to* **K*****8**.
Define $\mathfrak{M}_K = \langle W, \prec, V\rangle$ as follows:

- *W is the set of all maximal consistent theories of the language in which K is formulated;*
- *$u \prec w$ iff $\tau(w) \subseteq \tau(u)$ and $u \notin \tau(w)$.*

  Here, τ is given by: $v \in \tau(w)$ iff $v \in W$ and there is some φ such that $K^*_\varphi \subseteq w$ and $\varphi \in v$.
- *$V(p, w) = 1$ iff $p \in w$.*

Then $\mathfrak{M}_K = \langle W, \prec, V\rangle$ is a model for K for which the following holds:

$$\psi \in K^*_\varphi \textit{ iff } \psi \in w \textit{ for every } w \textit{ such that } w \textit{ is } \prec\textit{-minimal in } [\![\varphi]\!].$$

This means that whenever $\psi \in K^*_\varphi$, the model $\mathfrak{M}_K$ verifies $\varphi \rightsquigarrow \psi$. This model is almost connected. Therefore, in view the observations we made in section 1 and 10, it follows that the AGM revision constraints endow $\rightsquigarrow$ with the logic $\mathbf{P}$ + ASP.[11]

One may be tempted to conclude from the above that revising ones beliefs by φ and making the counterfactual assumption *if it had been the case that* φ amount to the same thing. However, even though these cognitive operations have much in common formally, there are huge differences between them. When you believe that φ is true and you try to imagine what would have been the case if φ had been false, you have to change your cognitive state, but it is it not the kind of change you would have to make if you were to discover that φ is *in fact* false. It is not a *correction.* Consider for example $\varphi =$ *Oswald killed Kennedy.* Supposing that Oswald had not killed Kennedy might make you think 'If Oswald had not killed Kennedy, Kennedy might still be alive'. If, however, at some point you were to find out that your belief that Oswald killed Kennedy is in fact wrong, and you had to revise your beliefs accordingly, it is very likely that after this revision you would still believe that Kennedy is dead.[12]

The rise of 'dynamic' versions of epistemic and doxastic logic have given new impetus to the study of of belief revision in the setting of modal logic. See van Benthem[74], Segerberg[68, 69] and Chapter 20 for further details.

BIBLIOGRAPHY

[1] C. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction functions and their associated revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

[2] H.M. Aldewereld, W. van der Hoek, and J.-J.Ch. Meyer. Rational teams: Logical aspects of multi-agent systems. In B. Dunin-Keplicz and R. Verbrugge, editors, *Proc. of the 1st Int. workshop on Formal Approaches to Multi-Agent Systems (FAMAS'03)*, pages 35–52, Warsaw, 2003. ACM Press.

[3] A. R. Anderson. A reduction of deontic logic to alethic modal logic. *Mind*, 67:100–103, 1958.

[4] N. Asher. Commonsense entailment: A conditional logic for some generics. In G. Crocco, L. Fari nas del Cerro, and A. Herzig, editors, *Conditionals: From Philosophy to Computer Science*, pages 103–145. Oxford University Press, Oxford, 1995.

[5] B. Bennett, C. Dixon, M. Fisher, E. Franconi, I. Horrocks, U. Hustadt, and M. de Rijke. Combinations of modal logics. *Artificial Intelligence Review*, 17(1):1–20, 2002.

[6] R. Blutner. *Normality* in update semantics. In M. Simons and T. Galloway, editors, *Proceedings from Semantics and Linguistic Theory V*, pages 19–36, Ithaca, New York, 1995. Cornell University.

[7] R.H. Bordini, M. Fisher, C. Pardavila, and M.J. Wooldridge. Modelchecking agentspeak. In *Proc. 2nd Int. Joint Conf. on Autonomous Agents and Multi Agent Systems (AAMAS'03)*, pages 409–416. ACM Press, 2003.

[8] M. E. Bratman. *Intentions, Plans and Practical Reason.* Harvard University Press, 1987.

[9] J. Broersen. Action negation and alternative reductions for dynamic deontic logics. *J. of Applied Logic*, 2(1):153–168, 2004.

[10] J. Broersen, M. Dastani, Z. Huang, J. Hulstijn, and L. van der Torre. Goal generation in the boid architecture. *Cognitive Science Quarterly*, 2(3-4):428–447, 2002.

[11] J. Broersen, M. Dastani, J. Hulstijn, and L. van der Torre. The boid architecture: Conflicts between beliefs, obligations, intentions and desires. In *Proc. 5th Int. Conf. on Autonomous Agents (AA2001*, pages 9–16, New York, 2001. ACM Press.

[12] J. Broersen, M. Dastani, and L. van der Torre. Resolving conflicts between beliefs, obligations, intentions and desires. In S. Benferhat and Ph. Besnard, editors, *Proc. ECSQARU'01*, pages 568–579, Berlin, 1997. Springer.

[13] J. Broersen, R.J. Wieringa, and J.-J. Ch. Meyer. A fixed-point characterization of a deontic logic of regular action. *Fundamenta Informaticae*, 48(2-3):107–128, 2001.

[11] An extensive discussion of the AGM theory of belief revision and its representation in conditional logic can be found in [36].

[12] See [66] for an insightful discussion of these points.

[14] J. P. Burgess. Quick completeness proofs for some logics of conditionals. *Notre Dame Journal of Formal Logic*, 22(1):76–84, 1981.
[15] J. Carmo and A.J.I. Jones. Deontic logic and contrary-to-duties. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic (2nd edition), vol. 8*, pages 203–279. Kluwer, Dordrecht, 2003.
[16] H.-N. Castañeda. The paradoxes of deontic logic: The simplest solution to all of them in one fell swoop. In R. Hilpinen, editor, *New Studies in Deontic Logic*, pages 37–85. Reidel, Dordrecht, 1981.
[17] B. F. Chellas. *Modal Logic: An Introduction*. Cambridge University Press, Cambridge, England, 1980.
[18] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(3):213–261, 1990.
[19] P. R. Cohen and H. J. Levesque. Teamwork. *Nous*, 24(4):487–512, 1991.
[20] M. de Rijke. A system of dynamic modal logic. *J. of Philosophical Logic*, 27:109–142, 1998.
[21] J. P. Delgrande. Conditional logics for defeasible logics. In Phillipe Besnard and Anthony Hunter, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems, Volume 2: Reasoning with Actual and Potential Contradictions*, pages 135–174. Kluwer Academic Publishers, Dordrecht, 1998.
[22] F.P.M. Dignum, D. Kinny, and E. Sonenberg. From desires, obligations and norms to goals. *Cognitive Science Quarterly*, 2(3-4):407–430, 2002.
[23] F.P.M. Dignum, J.-J. Ch. Meyer, and R.J. Wieringa. Contextual permission: A solution to the free choice paradox. In A.J. Jones and M. Sergot, editors, *Proc. 2nd Int. Workshop on Deontic Logic in Computer Science (DEON'94)*, pages 107–135, Oslo, 1994. Tano A.S.
[24] F.P.M. Dignum, J.-J. Ch. Meyer, and R.J. Wieringa. A dynamic logic for reasoning about sub-ideal states. In J. Breuker, editor, *Proc. ECAI'94 Workshop "Artificial Normative Reasoning")*, pages 79–92, Amsterdam, 1994.
[25] F.P.M. Dignum, J.-J. Ch. Meyer, and R.J. Wieringa. Free choice and contextually permitted actions. *Studia Logica*, 57(1):193–220, 1996.
[26] F.P.M. Dignum and J.-J.Ch. Meyer. Negations of transactions and their use in the specification of dynamic and deontic integrity constraints. In M.Z. Kwiatkowska, M.W. Shields, and R.M. Thomas, editors, *Semantics for Concurrency, Leicester 1990*, pages 61–80. Springer-Verlag, London/Berlin, 1990.
[27] F.P.M. Dignum, D. Morley, E.A. Sonenberg, and L. Cavedon. Towards socially sophisticated bdi agents. In M. Tokoro, editor, *Proc. 4th Int. Conf. on Multi-Agent Systems (ICMAS-2000)*, pages 111–118, Boston, MA, 1996.
[28] F.P.M. Dignum and B. van Linder. Modelling social agents: Communication as actions. In M. Wooldridge, J. Müller, and N. Jennings, editors, *Intelligent Agents III: Agents Theories, Architectures, and Languages (ATAL-96)*, pages 205–218, Berlin, 1997. Springer.
[29] B. Dunin-Kęplicz and R. Verbrugge. Collective intentions. *Fundamenta Informaticae*, 51(3):271–295, 2002.
[30] M. Esteva, J.A. Padget, and C. Sierra. Formalizing a language for institutions and norms. In J.-J. Ch. Meyer and M. Tambe, editors, *Intelligent Agents VIII*, pages 348–366. Springer, Berlin, 1984.
[31] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge*. The MIT Press, Cambridge MA, 1995.
[32] K. Fine. Critical review of D. Lewis's 'Counterfactuals'. *Mind*, 84:451–458, 1975.
[33] M. Fischer. A survey of concurrent metatem – the language and its applications. In D.M. Gabbay and H.J. Ohlbach, editors, *Temporal Logic – Proceedings of the First International Conference*, pages 480–505, Berlin, 1994. Springer-Verlag.
[34] D. Gabbay. Theoretical foundations for non-monotonic reasoning in expert systems. In K. Apt, editor, *Proceedings of the NATO Advanced Study Institute on Logics and Models of Concurrent Systems*, pages 439–457. Springer-Verlag, 1985.
[35] P. Gärdenfors. *Knowledge in Flux*. The MIT Press, Cambridge, Massachusetts, 1988.
[36] P. Gärdenfors and H. Rott. Belief revision. In J.A. Robinson D. Gabbay, C.J. Hogger, editor, *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 4*, pages 35–132. Oxford UNiversity Press, 1996.
[37] G. Gargov and S. Passy. A note on boolean modal logic. In P. Petkov, editor, *Mathematical Logic, Proc. of Heyting'88*, pages 311–321. Plenum Press, 2001.
[38] M.L. Ginsberg. Counterfactuals. *Artificial Intelligence*, 30:35–79, 1986.
[39] D. Grossi, J.-J. Ch. Meyer, and F.P.M. Dignum. Modal logic investigations in the semantics of counts-as. In *Proc. 10th Int. Conf. on AI & Law (ICAIL'05)*, pages 1–9. ACM, 2005.
[40] A. Grove. Two modellings for theory change. *Journal of Philosophical Logic*, 17:157–170, 1988.
[41] J. Halpern, N. Friedman, and D. Koller. First-order conditional logic for default reasoning revisited. *ACM Transactions on Programming Languages and Systems*, 1(2):175–207, 2000.
[42] D. Harel. *First-Order Dynamic Logic*. Springer, Berlin, 1979.

[43] D. Harel. Dynamic logic. In D.Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic, vol. 2: Extensions of Classical Logic*, pages 497–604. D. Reidel Publishing Co., Dordrecht, 1984.
[44] E. Hemaspaandra. Complexity transfer for modal logic. In *Proc. IEEE Symp. on Logic in Computer Science (LICS'94)*, pages 164–173, Los Alamitos, 1994. IEEE Computer Society Press.
[45] K.V. Hindriks, F.S. de Boer, W. van der Hoek, and J.-J. Ch. Meyer. Agent programming in 3apl. *Int. J. of Autonomous Agents and Multi-Agent Systems*, 2(4):357–401, 1999.
[46] U. Hustadt, C. Dixon, R.A. Schmidt, M. Fisher, J.-J. Ch. Meyer, and W. van der Hoek. Verification within the karo agent theory. In J.L. Rash, C.A. Rouff, W. Truszkowski, D. Gordon, and M.G. Hinchey, editors, *(Proc. First Goddard Workshop on) Formal Approaches to Agent-Based Systems (FAABS 2000)*, pages 33–47, Berlin, 2001. Springer.
[47] A.J.I. Jones and M. Sergot. A formal characterization of —institutional power. *Journal of the IGPL*, 3:429–445, 1996.
[48] A. Kratzer. Partition and revision: the semantics of counterfactuals. *Journal of Philosophical Logic*, 10:242–258, 1981.
[49] A. Kratzer. An investigation of the lumps of thought. *Linguistics and Philosophy*, 87(1):3–27, 1989.
[50] S. Kraus and D. Lehmann. Knowledge, belief, and time. *Theoretical Computer Science*, 58:155–274, 1988.
[51] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 14(1):167–207, 1990.
[52] D. Lewis. *Counterfactuals*. Basil Blackwell, Oxford, 1973.
[53] D. Lewis. Counterfactual dependence and time's arrow. *Noûs*, 13:455–476, 1979.
[54] J.-J. Ch. Meyer. Modal epistemic and doxastic logic. In (D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic (2nd edition), vol. 10*, pages 1–38. KLuwer, Dordrecht, 2003.
[55] J.-J. Ch. Meyer. Reasoning about emotional agents. In R. L. de Mántaras and L. Saitta, editors, *Proc. 16th Eur. Conf. on Artif. Intell. (ECAI'04))*, pages 303–307. IOS Press, 2004.
[56] J.-J. Ch. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science*. Cambridge University Press, Cambridge, 1995.
[57] J.-J. Ch. Meyer, W. van der Hoek, and B. van Linder. A logical approach to the dynamics of commitments. *Artificial Intelligence*, 113:1–40, 1999.
[58] J.-J. Ch. Meyer, R.J. Wieringa, and F.P.M. Dignum. The role of deontic logic in the specification of information systems. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, pages 71–115. Kluwer, Boston/Dordrecht, 1998.
[59] J.-J.Ch. Meyer. A different approach to deontic logic: Deontic logic viewed as a variant of dynamic logic. *Notre Dame J. of Formal Logic*, 29(1):109–136, 1988.
[60] D. Nute. *Defeasible Deontic Logic*. Kluwer, Dordrecht, 1997.
[61] J. Pollock. *Subjunctive Reasoning*. Reidel, Dordrecht, 1976.
[62] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In J. F. Allen, R. Fikes, and E. Sandewall, editors, *KR'91: Principles of Knowledge Representation and Reasoning*, pages 473–484. Morgan Kaufmann, San Mateo, California, 1991.
[63] A.S. Rao. Agentspeak(l): Bdi agents speak out in a logical computable language. In W. van der Velde and J. Perram, editors, *Agents Breaking Away*, pages 42–55. Springer, Berlin, 1996.
[64] A.S. Rao and M.P. Georgeff. Decision procedures for bdi logics. *J. of Logic and Computation*, 8(3):293–344, 1998.
[65] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1–2):81–132, 1980.
[66] H. Rott. Moody conditionals: Hamburgers, switches, and the tragic death of an american president. In Jelle Gerbrandy, Maarten Marx, Maarten de Rijke, and Yde Venema, editors, *JFAK. Essays dedicated to Johan van Benthem on the occasion of his 50th birthday*, pages 98–112. Amsterdam University Press, Amsterdam, 1999.
[67] E. Sandewall and Y. Shoham. Nonmonotonic temporal reasoning. In D. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming Vol.4 (Epistemic and Temporal Reasoning*, pages 439–498. Oxford University Press, Oxford, 1994.
[68] K. Segerberg. Belief revision from the point of view of doxastic logic. *Bulletin of the IGPL*, 3(4):535–553, 1995.
[69] K. Segerberg. The basic dynamic doxastic logic of agm. In M.-A. Williams and H. Rott, editors, *Frontiers in belief revision*, pages 57–84, Dordrecht, 2001. Kluwer.
[70] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1):51–92, 1993.
[71] M.P. Singh. The intentions of teams: Team structure, endodeixis, and exodeixis. In H. Prade, editor, *Proc. 13th Eur. Conf. on Artif. Intell. (ECAI'98))*, pages 303–307, Chichester, 1998. Wiley.
[72] R Stalnaker. A theory of conditionals. In Nicholas Rescher, editor, *Studies in Logical Theory*, pages 98–112. Basil Blackwell, Oxford, 1968.
[73] P. Tichy. A counterexample to the stalnaker-lewis analysis of counterfactuals. *Philosophical Studies*, 29:271–273, 1976.

[74] J. van Benthem. Semantic parallels in natural language and computation. In H.-D. Ebbinghaus et al., editor, *Logic Colloquium '87*, pages 331–375. North Holland, Amsterdam, 1989.
[75] W. van der Hoek. Systems for knowledge and belief. *Journal of Logic and Computation*, 33:173–195, 1993.
[76] W. van der Hoek, J.-J. Ch. Meyer, and J.W. van Schagen. Formalizing potential of agents: The karo framework revisited. In S. Kaufmann M. Faller and M. Pauly, editors, *Formalizing the Dynamics of Information*, pages 51–67. CSLI Publications, Stanford CA, 2000.
[77] W. van der Hoek, B. van Linder, and J.-J. Ch. Meyer. An integrated modal approach to rational agents. In M. Wooldridge and A. Rao, editors, *Foundations of Rational Agency*, pages 133–168. Kluwer, Dordrecht, 1998.
[78] R. van der Meyden. The dynamic logic of permission. *J. of Logic and Computation*, 6(3):465–479, 1996.
[79] B. van Linder. *Modal Logics for Rational agents*. D.phil thesis., Utrecht University, Utrecht, 1996.
[80] B. van Linder, W. van der Hoek, and J.-J. Ch. Meyer. Actions that make you change your mind: Belief revision in an agent-oriented setting. In H. Wansing, editor, *Knowledge and Belief in Philosophy and Artificial Intelligence*, pages 103–146. Akademie Verlag, Berlin, 1995.
[81] B. van Linder, W. van der Hoek, and J.-J. Ch. Meyer. Seeing is believing (and so are hearing and jumping). *Journal of Logic, Language and Information*, 6:33–61, 1997.
[82] M.B. van Riemsdijk, F.S. de Boer, and J.-J. Ch. Meyer. Dynamic logic for plan revision in intelligent agents. In J. Leite and P. Torroni, editors, *Proceedings CLIMA V (5th Int. Workshop on Computaional Logic in Multi-Agent Systems)*, pages 196–211, 2001.
[83] F. Veltman. Prejudices, presuppositions, and the theory of counterfactuals. In J. Groenendijk and M. Stokhof, editors, *Amsterdam Papers in Formal Grammar. Proceedings of the 1st Amsterdam Colloquium*, pages 248–281. University of Amsterdam, 1976.
[84] F. Veltman. *Logics for Conditionals*. Ph.D. dissertation, University of Amsterdam, Amsterdam, 1985.
[85] F. Veltman. Defaults in update semantics. *Journal of Philosophical Logic*, 25:221–261, 1996.
[86] F. Veltman. Making counterfactual assumptions. *Journal of Semantics*, 22:159–1880, 2005.
[87] G. H. von Wright. Deontic logic. *Mind*, 60(237):1–15, 1951.
[88] G. H. von Wright. A new system of deontic logic. *Danish Yearbook of Philosophy*, 1, 1964.
[89] F. Voorbraak. The logic of objective knowledge and rational belief. In Jan van Eijck, editor, *Logics in AI (Proceedings of JELIA '90)*, pages 499–516, Berlin, 1991. Springer.
[90] F. Voorbraak. *As Far as I Know: Epistemic Logic and Uncertainty*. Ph.D. dissertation, Utrecht University, Utrecht, 1993.
[91] R. Wieringa, J.-J.Ch. Meyer, and H. Weigand. Specifying dynamic and deontic constraints. *Data & Knowledge Engineering*, 4(2):157–190, 1989.
[92] R.J. Wieringa and J.-J. Ch. Meyer. Actors, actions, and initiative in normative system specification. *Annals of Mathematics and Artificial Intelligence*, 7:289–346, 1993.
[93] M. J. Wooldridge. *Reasoning about Rational Agents*. The MIT Press, Cambridge, MA, 2000.
[94] M.J. Wooldridge and P. Ciancarini. Agent-oriented software engineering: The state of the art. In P. Ciancarini and M.J. Wooldridge, editors, *Agent-Oriented Software Engineering*, pages 1–28. Springer, Berlin, 2001.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

19

APPLICATIONS OF MODAL LOGIC IN LINGUISTICS

Lawrence S. Moss and Hans-Jörg Tiede

1 Introduction . . . 1031
2 Semantics . . . 1033
2.1 Possible worlds in semantics . . . 1033
2.2 Specific contributions: an overview . . . 1036
2.3 Intensionality . . . 1037
2.4 Propositional attitudes . . . 1038
2.5 Conditionals . . . 1039
2.6 Time and tense . . . 1040
2.7 The reference time . . . 1043
2.8 Temporal reference and hybrid logic . . . 1044
2.9 A note on multidimensionality . . . 1046
2.10 Problems and prospects . . . 1046
2.11 Montague semantics . . . 1047
3 Syntax . . . 1054
3.1 Mathematical linguistics . . . 1054
3.2 Preliminary: logics of strings . . . 1055
3.3 Logics of trees . . . 1061
3.4 Assessment: why modal logic for syntax and which one? . . . 1072
4 Conclusion and open problems . . . 1072

1 INTRODUCTION

That logic and language are closely related is almost true by definition. Logic is concerned with the study of valid inferences in arguments, and these are most commonly defined in terms of truth in models. Symbolic logic studies formal languages (logics) as models of certain aspects of natural languages, such as quantification, while abstracting away from certain other aspects of natural languages, such as ambiguity, as models typically do. Linguistics studies the structure of natural languages as well as the relation of language to other areas of cognitive science. The roles that logic in general, and modal logic in particular, play in linguistics are quite varied, as we shall see.

In linguistic semantics, logic is used to formalize, or interpret, an *object language*. We take as given that we want to study the semantics of some natural language, and in this

chapter the language that we shall deal with is English. Above all else, we would like to directly interpret English sentences in some formally specified model. So even at this point we can see some connection to the Kripke semantics of modal operators: just as all of the other phenomena in this "applied" section of the handbook have been modeled with mathematical structures involving possible worlds, so too have these been used in semantic applications. For example, all manner of linguistic phenomena involving time have led to proposals for using the models from temporal logic. More generally, the main models of all types of *intensional* phenomena are closely related to the models in modal logic.

But so far we have only considered the matter of interpreting natural language directly. Usually, this is difficult or even impossible. (For example, consider the famous quantifier scope ambiguities in sentences like *every handbook has a famous editor*. The ambiguity is neatly expressed in logical notation as $\exists\forall$ vs $\forall\exists$: is there one person, let's call him Dov, who edits all the handbooks, or is it merely that every handbook has some editor or other? One lesson to take from such ambiguities is that it is impossible to associate a function from (English $\times$ models) to truth values in a way that respects our intuitions.) So one way or another, we translate natural language to some artificial language and then interpret that other language, in such a way that ambiguous sentences will be translated into multiple logical formulas. And here is a second place modal logic comes in: the language of higher-order modal logic has been used extensively to drive this translation process, as we shall see when we discuss Montague semantics.

We next turn to syntax, a field in which one finds several different uses of logic. There are syntactic frameworks which are heavily proof-theoretic, so the question of whether a given string is a sentence or not boils down to whether a related (formal) sentence is a theorem in some logical system. This proof-theoretic move is especially prominent in categorial grammar. Another quite different use of logic is as a *meta language* in which one formalizes a linguistic formalism declaratively. This is the move of *model theoretic syntax*, a research program we consider in depth in the second half our chapter. This application relates logic to linguistics in the same way that logic can be applied to formalize theories of other sciences, like set theory. However, the aims of this formalization are somewhat different from those of other areas, since model theoretic syntax is particularly interested in using *decidable* logics for this formalization so that matters can be implemented. This is of course one of the reasons why modal logics are attractive in this context, although much of the focus has been on monadic second-order logic of trees, which is decidable as well.

Applications of logic in linguistics have traditionally not been too concerned with meta-results. The main uses of modal logic in semantics are independent from the main concerns of modal logicians: completeness and correspondence. We are not aware of any serious application of the basic theory of modal logic in semantics, let alone the advanced theory that is showcased in various chapters of this handbook. The only exception is definability theory, interest in which is motivated by trying to find a logic for linguistic applications that has the right kind of "expressiveness." For example, the fact that *most A are B* is not first-order definable is of some importance for semantics. On the other side, the application of logic in syntax has led to more applications of sophisticated meta-results, for example proof theoretical results like cut-elimination or normalization in categorial grammar. It is interesting to note that definability is also of importance in model theoretic syntax, due to its relation to descriptive complexity theory. A re-

lated point: because so many current syntactic frameworks are designed with a hope of implementation, sharper theoretical results about them are called for.

In this chapter, we only survey applications of modal logic to the syntax and semantics of natural languages. We concentrate on these two applications because of the historical importance of modal logic in the development of natural language semantics and because of the significance of model theoretic syntax in current research in mathematical linguistics. There are many areas of applications of logic in linguistics that we do not mention, some of which are surveyed in the *Handbook of Logic & Language* [4].

2 SEMANTICS

Linguistic semantics studies meaning in natural languages. The central assumption of current semantic theory is that meaning should be studied model theoretically, in the same way that semantics of logics are studied. Thus, the study of *meaning* is tied to the concept of *truth*. Of course, there are other ways to pursue the project of understanding meaning, most notably to tie it to *action* in some way. As it happens, for some purposes possible worlds semantics is even better for this second purpose than for the first; see, for example, [69].

The interpretation of logical formulas usually involves the interpretation of subformulas in some systematic fashion. For instance, in propositional logic we have interpretational clauses like

$$[\![\varphi \wedge \psi]\!] = [\![\wedge]\!]([\![\varphi]\!], [\![\psi]\!])$$

where $[\![\wedge]\!]$ is the boolean *and* function. The methodological principle that stipulates that all interpretations of complex expressions should involve the interpretations of its parts is called the "principle of compositionality," and it plays a central role in linguistic semantics. Whether that principle is in fact a meaningful restriction on semantic theory or whether it is vacuous is a point of ongoing debate. For one source that discusses the matter at length, see Janssen [41].

Since natural language semantics applies model theoretic methods, the role of modal logic in this context involves the application of possible worlds semantics to natural languages, mainly to model *intensional* phenomena. However, in order to follow the principle of compositionality uniformly, the meanings of some expressions are modeled using higher-order logic. Thus, the most influential, systematic application of modal logic to linguistic semantics, usually referred to as *Montague semantics* after its founder, involved *higher-order intensional logic.* Although Montague's application of higher-order intensional logic to natural language semantics yielded many important results, almost all of the contemporary research is concerned with finding suitable alternatives to this framework. Many of these are surveyed in the *Handbook of Logic and Language* [4]. Another handbook in this area, with a more empirical and linguistic, as opposed to theoretical and logical, slant is the *Handbook of Contemporary Semantic Theory* [57]. There are many introductory textbooks in linguistic semantics, including [15, 22, 36].

2.1 Possible worlds in semantics

The major use of modal logic in semantics stems from possible worlds semantics. Indeed, this is the only kind of application we are considering in this chapter.

This is a good point to make a comment which relates to the place of this chapter in the overall handbook, and also one relating to current practice in the field of modal logic. One of the main subtexts in this volume is that research on modal logic has much to contribute to other areas. So the volume downplays the problematic points of possible worlds semantics by emphasizing topics in modal logic which are interest in areas far removed from those problems. In other words, one can put aside ontological worries (as one would like to do in any mathematical study) because in the kind of transition-system models emphasized in the study, these worries are not relevant. This kind of move is not appropriate for semantics: on the one side, problems about the status of possible worlds come up quicker and they persist; we shall shortly see an example. On the other side, there are few, if any, technical matters of interest in semantics. It is essentially all a matter of studying data from language, proposing treatments that use possible worlds, describing informal models related to the phenomena or the treatments, and occasionally working out the semantics of one or another formal logical language. In this section, we are mainly trying to provide a reader who is conversant with modal logic a feeling for what goes on in semantics.

Here is an example that motivates the use of possible worlds in semantics, taken from McCawley [64]. In a normative English sentence, two uses of a first person pronoun (*I/me/myself*) must be coreferential. And if the sentence has both a first person subject and object, the object must be the reflexive pronoun *myself*. So one cannot say *I kissed me* but instead must have *I kissed myself*. The only exception to this, and this is heart of the matter, is that "Multiple references for first-person pronouns arise when the sentence alludes to an alternative world in which the speaker ... is presented as experiencing something from someone else's vantage point" (McCawley [64]). For example,

(1) I dreamed that I was Brigitte Bardot and that I kissed me.

This would not mean the same thing as *I dreamed that I was Brigitte Bardot and that I kissed myself*. Getting back to (1), it shows what appears to be a hard-and-fast syntactic rule has to be understood in essentially semantic terms. (This is not as surprising as it might at first be: try formulating a principle of reflexive pronouns without using the semantic concepts of *subject* and *object*.)

(1) also shows that in some implicit sense speakers refer to "dream worlds" and more generally to alternative worlds of other kinds, or alternative ways this world could be. Note that the status of who *I* and *me* are in the dream world is problematic, but we shall not delve into this. The point is that if one wants to construct a formal semantics for (1) using Kripke models, then *prima facie* one would want to use worlds: a world where the speaker has a dream, and a world that represents what is happening in that dream. Note as well that what happens in dreams might be logically inconsistent, so Kripke models as one standardly finds them in modal logic are not going to be sufficient for representations of this kind. But they are a useful first step. Indeed, practicing semanticists have found the informal talk about possible worlds to be convenient and motivating. Like contemporary modal logicians, they are usually not interested in, or bothered by, worries about whether possible worlds are real. But again, the difference is that for modal logicians, the worries go away precisely because they tend to avoid modeling anything like an imaginary world, something which is evidently of linguistic interest.

For another example of why semanticists want to think in terms of possible worlds,

consider the following contrasting sentences:

(2) i. It's certain that you'll find a job, and it's conceivable that it will be a good-paying one.

ii. ??It's conceivable that you'll find a job, and it's certain that it will be a good-paying one.

This example is from McCawley [64] in a section entitled "'World-Creating' Predicates." (Incidentally, the quotes here are his, betraying already a certain discomfort with either the notion or the terminology. In any case, we shall expand on just this point below. But the terminology again shows an embrace of possible worlds as well.) The question marks in (2ii) indicates a semantic anomaly. That is, what appears strange in (2ii) is not due to syntactic ill-formedness: from (2i) and the fact that *certain* and *conceivable* both take sentence complements, one would expect (2ii) to be grammatical. So one of the goals of any analysis would be a principled explanation of the different acceptability judgments between (2i) and (2ii).

We encourage readers who are not familiar with semantics to attempt a translation into any logical language of (2i), and also to draw pictures of Kripke models to explain their intuitions. Incidentally, in both (2i) and (2ii), we are concerned with the "non-referential" interpretations: there is no specific job that Gladys is looking for.

Since the sentences in (2) are in the future, a representation should have at least two worlds: the present world, in which (presumably) you do not have a job but are seeking one, and at least one future world. Thus we are inclined to model (2i), say, by having one actual world, w and *many* alternative worlds for the relatively-near future, each with the property that you find some job or other in it. This is quite typical for semantic analysis: if one is going to use worlds to represent alternatives which are only partially specified, then very quickly one must consider many alternatives. Frequently there will be infinitely many. It is the second half of the sentence which gives more trouble; it seems to require that in some (or perhaps some significant proportion of) the successors of w, your job is a good-paying one. And even with this sketch of a representation, we have not learned the lesson that (2) teaches. The point is that a use of *it* in the second part of (2i) is dependent on the existence of a job. If the existence of a job is in doubt, as it is in the first part of (2ii), then it is infelicitous to use *it* to refer to one later. This kind of reasoning could be fleshed out in a fuller analysis in several ways. One would be to use a theory of presupposition. Another would be to ground the whole discussion in Discourse Representation Theory (DRT) [43] or some "dynamic" theory which has enough theoretical apparatus around to talk about different occurrences of pronouns like *it*. In any case, a DRT analysis of the sentences in (2) would most likely use possible worlds at the very least.

One last comment: from more sophisticated examples, such as ... *it is more likely than not that it will be a good-paying one*, we see that complex relations between worlds are going to be the norm rather than the exception. These relations can involve additional structure, as in this probabilistic setting, or various notions of nearness (as we find in treatments of conditionals).

2.2 Specific contributions: an overview

In an assessment of the importance of possible worlds semantics for linguistics, Partee [74] highlights the following six areas:

1. The identification of propositions with sets of possible worlds.
2. The analysis of intensional phenomena with functions from possible worlds to their extensions.
3. The semantics of propositional attitudes.
4. The semantics of conditionals.
5. The semantics of questions and the pragmatics of the question-answer relations.
6. Pragmatics in general, and presuppositions in particular.

Beginning with this subsection, we shall explore several of these contribution areas in detail. Some of these topics are not treated in our chapter: we won't have much to say about questions and answers or pragmatics. The prevalent view of the role of pragmatics in linguistics is that it is the part of semantics that is concerned with the *context-dependent* meaning of linguistic expressions. In the narrowest sense, pragmatics is concerned with the interpretation of indexical or deictic expressions, like personal pronouns, and that is the sense in which the term was used by Montague [99]. Since then, the subject matter of pragmatics has been extended to include many of the topics discussed by ordinary language philosophers like Austin, Grice, Strawson, and Searle. Thus, pragmatics now includes topics such as implicature, presupposition, and speech acts. There is also some overlap between pragmatics and sociolinguistics, though that overlap has little to do with modal logic.

Of the subjects of pragmatics mentioned above, primarily indexicals and presuppositions have been analyzed using possible worlds semantics, although Posner [78] reconstructs communicative actions and ultimately speech act theory in terms of suitable iterations of modal operators for believing, causing, and intending.

Concerning work related to questions and answers, we only mention one quite recent reference, the dissertation Murakami [68]. This proposes an analysis of notions like *complete and just complete answer to a question* based on the modal logic of *partitions*.

With respect to the first two items listed by Partee, one set of notions worth keeping in mind goes back at least to Carnap [14]. He emphasized the distinction between extension and intension, and linked these to particular syntactic items as follows:

Expression	Intension	Extension
sentence	proposition	truth-value
predicate	property	set
individual term	individual concept	individual.

Note that the identification of propositions with sets of possible worlds is a special case of this analysis, since sets of possible worlds are essentially the same as functions from possible worlds to the set of truth values. In essence, this identification of propositions with sets amounts to the imposition of a boolean algebra structure on the set of propositions. This kind of structure needs to be supplemented with accessibility relations or other more typically modal structures to be relevant to our discussion.

2.3 Intensionality

We have yet to discuss intensionality in a general way, and now is the time to do just that. It is relatively uncontroversial that some linguistic expressions refer to objects. For instance, names refer to the object that they name. Frege [29] postulated that there is another dimension to "meaning" other than reference in order to explain the apparent difference between statements of the form $a = b$ and $a = a$. While the latter is true *a priori*, the former, when true, typically requires some kind of observation. The difference between these two statements, according to Frege, is that, if $a = b$ is true, a and b have the same *reference* (*Bedeutung*), but different *sense* (*Sinn*). However, Frege did not give a formal definition of "sense." Carnap [14] used the concepts of *extension* and *intension* as a model of reference and sense, which is one of the first and most influential applications of modal logic to natural language semantics. Carnap, however, used *state descriptions*, maximally consistent sets of literals, instead of Kripke models for the semantics of modal logic. Montague is credited with bringing together Carnap's analysis of sense and reference with Kripke's possible world semantics. (Incidentally, a different analysis of *sense* along algorithmic lines has been suggested in recent years by Moschovakis; see [67].)

David Lewis [59] expressed the motivation for this approach to natural language semantics particularly well:

> In order to say what meaning *is*, we may first ask what a meaning *does* and then find something that does that. [...] It is the meaning which determines how the extension depends on the combination of relevant factors. What sort of things determine how something depends on something else? *Functions*, of course [...]

Thus, *intensions* are defined by Lewis to be functions from possible worlds (and possibly other indices, which are the *relevant factors*, above) to extensions. This is the central idea behind the possible world analysis of intensionality. So far, we have only mentioned the extensions of names: individuals. What about the extensions of other kinds of expressions? It has been part of the Fregean orthodoxy to consider the extension of a sentence to be its truth value. Since intensions are functions from possible worlds to extensions, the intensions of sentences are functions from possible worlds to truth values, or simply sets of possible worlds; i.e., those possible worlds in which the sentence is true. The intensions of sentences are also called "propositions." As Partee [74] points out, this analysis of the meaning of sentences gives a good approximation to the notion of "synonymity," since two sentences that have the same intensions are true in exactly the same possible worlds.

Other, classical, examples of intensional phenomena include intensional transitive verbs, like *seek*, and propositional attitude verbs, like *believe*. Interacting with both of these are the *de re* and *de dicto* distinctions. To illustrate this distinction, consider the following sentence, and note the intensional transitive verb: *Barney wants to drive the fastest car in town.* One reading of this sentence is where there is a specific car, say c, and Barney wants to drive c. (But the fact that c is the fastest car in town is not germane to Barney's wish: he just wants to drive c.) This is the *de re* reading. The *de dicto* reading is where Barney wishes to drive whatever car is the fastest in town; so if that description were to change referents over time, then Barney's desire would also change.

Before moving on, we should mention that the entire treatment of intensionality via possible worlds semantics is not universally accepted in semantics by any means. A good source for some criticisms is John Perry's side [76] of a discussion with Barbara Partee. In other areas as well, one has the feeling that the whole application of possible worlds in semantics is, as one prominent semanticist privately told one of us, a "counterfactual exercise": even though possible worlds semantics are the community's standard and the best thing known, many researchers believe that in the long run they cannot succeed at everything they are being applied to.

2.4 Propositional attitudes

The phenomena of interest in the context of propositional attitudes are belief and knowledge, and also the *root modals* like *can*, *may*, and *should*. The main lines of the standard treatments are probably closest to the heart of a semantically-oriented modal logician: one takes a space of worlds which is equipped with a relation corresponding to each attitude or modality of interest. Then one defines semantics for the attitudes themselves as modal operators in the expected way, by quantifying using the accessibility relations. Modal logicians will also recognize the parallel to the algebraic semantics of modal logic; see Chapter 6. The point is that by moving to the power set algebra of the set of worlds of a model is like moving to the space of *intensions*. So the propositional attitudes turn into operators on the intensional rather than extensional level. This is a two-edged sword: on the one hand, it allows us to explain why statements of identity are not preserved in modal contexts. But the down side is the problem of logical omniscience: logically necessary propositions wind up as being known by everyone at every point. So exactly the same advantages and disadvantages come up as in the theory of knowledge.

Here is one textbook treatment of the basics of propositional attitudes, following the final chapter of Heim and Kratzer [36]. This chapter is called a "first step" on the way to an intensional semantics, and the authors emphasize, and close with, the limitations of their work. The point for us concerns the treatment of the attitude verbs such as *know* and *believe*. The way things work syntactically, attitude verbs take sentences as their arguments; a verb phrase then results. So their categorial type (see Section 2.11 below) would be VP/S, and so their semantics is a function from propositions (i.e., functions from worlds to truth values) to VP meanings (here functions from individuals x to truth values). The semantics is then given by

(3) $[\![believes]\!] =$
$\lambda w.\lambda p.[\lambda x.(\forall w')(\text{if } w' \text{ is belief-consistent with } w \text{ for } x, \text{ then } p(w') = 1)].$

"Belief consistent" here a relation on worlds defined as follows: we say that w' is belief consistent with w for (person) x if all of x's beliefs in w are true in w'. (So in effect this treatment does not work on a set Kripke model whose accessibility relation is up to the semanticist to specify, but rather that the accessibility relation is *given* in terms of what we are calling belief consistency here.)

To illustrate this, we consider a sentence of the form *Mary believes S*, where S is another sentence. Overall principles of compositionality insure that for all worlds w,

$$[\![\textit{Mary believes S}]\!]_w = [\![\textit{believes S}]\!]_w(\textit{Mary}).$$

The definitions in the semantics are set up so that the following holds:

(4) $[\![believes\ S]\!]_w(Mary) = [\![believes]\!]_w(\lambda w'.[\![S]\!]_w(Mary))$.

At this point we apply the general definition of $[\![believes]\!]$ from (3). We see that *Mary believes S* is true in a world w just in case for all worlds w' which are belief consistent with w for Mary, S is true in w'. So in this way we reconstruct the semantics of the attitudes which would be expected from the Kripke semantics of modal logic. We shall discuss how the calculation in (4) works when we turn to Montague grammar in Section 2.11.

Before returning to a discussion of the modalities in language, here is another point on the treatment of belief in linguistics. McCawley [64] suggests a departure from what modal logicians might expect concerning belief when he writes, "Belief worlds may even conform to a different version of logic than the real world is taken to be subjected to; such worlds would be appropriate devices for analyzing such sentences as those in which an adherent of standard logic attributes beliefs to an intuitionist." But he also holds also that "one has a single set of beliefs at a time (possibly inconsistent beliefs, but a single set nonetheless)." So this seems to suggest that belief worlds might be *paraconsistent* in some sense. But later, in connection with wishes, he is of the opinion that "It will probably be clearest if one simply avoids terms such as 'wish world,' which misleadingly suggest that there is a single system of wishes whose simultaneous fulfillment is at issue, and instead use circumlocutions to say that a particular world corresponds to a particular wish. ... These worlds may serve as reference worlds for other worlds that correspond to, say, the fulfillment of wishes, hopes, and so forth, that are contingent on the fulfillment of a given wish."

In any case, much of the linguistic discussion is not about these kinds of points, but rather questions of reference and presupposition. For example, here are sentences from McCawley [64], page 426:

(5)
i. Arthur thinks that <u>a unicorn</u> has been eating his roses. He hopes he can catch <u>it</u>.
ii. Arthur denies that <u>a unicorn</u> has been eating his roses. ??He hopes he can catch <u>it</u>.

The underlined *it* in the first sentence refers to the unicorn in the preceding sentence. Actually, *a unicorn* is best understood non-referentially here; there is no particular unicorn which Arthur is thinking about, just some-unicorn-or-other. The point is that it is a property of *think* that it allows nonreferential NP's in its complement to be the antecedents of later pronouns (the subsequent *it*). In contrast, *deny* does not have this property. This is why the second sentence in (2 ii) is anomalous. This last example is intended to be more typical of the linguist's concerns than the previous paragraph on ontological points.

2.5 Conditionals

Modal-type notions are of central interest in work on conditionals, following Lewis [60] and Stalnaker [90]. The idea here is to analyze counterfactual conditionals (those whose antecedents are known or taken to be false) using a semantics that comes with some extra apparatus or other. One should see Chapter 18 for an extensive treatment of the logical systems that come from the natural semantics of counterfactual conditionals. But counterfactuals are only one type of conditionals, and another important type are

indicative conditionals (where the antecedent is true). See von Kutschera [106] for a proposal on indicative conditionals related to, but different from, the standard treatment of counterfactuals.

As it happens, most contemporary work in semantics does not use the Lewis-Stalnaker semantics but instead works with elaborations based on it. Probably the main proposal in the area is due to Kratzer [55]. Her work allows one to work with the combination of conditionals and modals, as in *If this is an article on linguistics, there must be examples from many languages.* Kratzer's semantics makes use of a "modal base"; this is basically a spelled out version of an accessibility relation. It also uses a three-placed similarity relation on worlds. It was later observed by Frank [28] and independently Zvolensky [107] that sentences of the form *If X, then it must be the case that X* came out automatically true in Kratzer's semantics. Modal logicians might find it interesting to note that a similar debate about the adequacy of modal semantics crops up in areas like deontic logic. Indeed, one of the lessons we learned in writing this chapter is exactly that similar questions about the adequacy of various semantic proposals coming from modal logic come up independently in different forms. From the point of view of *applied modal logic* it would clearly be of value for people to pay close attention to points like this, in order to make theoretical contributions that could be appreciated by people in different fields.

2.6 Time and tense

Another semantic area where ideas of possible worlds semantics are put to use concerns time and tense in natural language. Our discussion of these issues is once again intended only as an invitation to this fascinating field. It is based largely on the survey of the area in Mark Steedman's draft textbook [91] and also on Dick Crouch's ESSLLI notes [21]. We have also again found McCawley's book [64] full of insightful examples and proposals; see Section 12.2 on Tense Logic.[1] An essential resource for researchers in this area is Robert I. Binnick's web site [5] entitled "The Project on Annotated Bibliography of Contemporary Research in Tense, Grammatical Aspect, Aktionsart, and Related Areas":

`http://www.scar.utoronto.ca/~binnick/TENSE/`.

In particular, the logic part of the site lists a large collection of papers relevant to the subject of this handbook chapter.

For readers of this handbook, perhaps the primary observation concerning the analysis of time in natural language is that the whole matter of *temporal ontology* is highly complicated and problematic. First of all, there are words, endings, and expressions which are *usually* used to indicate past, present, and future time references. But even these have exceptions. For example, *ing* usually indicates a present tense, but in examples like *the editors are calling Larry tomorrow to complain that his paper contained a lot of misleading remarks*, the word *tomorrow* changes this to a future time reference. Yet

[1]Indeed, at various places in this chapter we have marveled at McCawley's use of the best logical tools available to him. He wrote "I teach courses on logic from a linguists point of view, taking a broad view of the subject matter of logic (logic has suffered 23 centuries of myopia, which I try to make up for) and giving full weight to linguistic considerations in revising (or replacing) existing systems of logic to maximize their contact with natural language syntax and linguistic semantics." [65] We therefore wonder what he as a linguist would have found useful in the exploding logical literature. Although we never met him personally, we would like to think that some of our comments about various connections and possible applications would have inspired him (and those who follow in his footsteps).

another point concerns *embedded tenses*; as our last example shows, it is not always straightforward even to interpret these constructions, let alone represent and analyze them. For another example along the same lines, *Sonia said that Rajiv liked to dance* should have the same meaning as *Sonia said "Rajiv likes to dance"*; the problem then is to account for this sameness. Finally, important temporal information is often absent from the surface forms. Consider

(6) John went to kindergarten with a bank president.

The intended meaning is that at some past time John went to kindergarten with some individual who would later become a president of a bank.

The first, and most basic, proposal for the representation of temporal phenomena is to add an explicit time parameter to propositional functions. So instead of a predicate like $alive(x)$ which indicates whether an object is alive or not, we might have $alive(x, t)$. Then one might want to translate various tense constructions into, say, a two-sorted first-order logic; the point is that one then has quantification over times and also a symbol $<$ for the relation of *preceding* on times. Then one can translate a future sentence like *Sonia will go* as

$$(\exists t > t_0)(go(Sonia, t)).$$

Note that there is a "now" time t_0; this can be taken to be either a constant or a variable.

However, this is usually not what is done. There are logical and also linguistic reasons for making other moves. In a comment directly related to this, Thomason [100] writes:

> Physics should have helped us realize that a temporal theory of a phenomenon X is, in general, more than a simple combination of two components: the statics of X and the ordered set of temporal instants. The case in which all functions from times to world-states are allowed is uninteresting; there are too many such functions, and the theory has not begun until we have begun to restrict them. And often the principles that emerge from the interaction of time with the phenomena seem new and surprising.

The new and surprising principles here are the interactions of tense and modality that Thomason discusses in his handbook article [100]. But mention of physics also raises the question of the structure of time. In the linguistic literature, the emphasis nearly always is on what might be called *linguistic time*, the common-sense notion that we want to tease out and model from "people on the street". It is not the notion that we would get from physics.

An alternative way to go is to take the basic sentences of language to be tenseless and then to add temporal modal operators P (for the past) and F (for the future). This is the basic move of Tense Logic, usually mentioned in connection with its main developer, Arthur Prior.

These are interpreted on linear orders $(L, <)$. The semantics is the standard one from temporal logic

$$(7)\quad \begin{array}{lll} l \models P\phi & \text{iff} & m \models \phi \text{ for some } m < l \\ l \models F\phi & \text{iff} & m \models \phi \text{ for some } m > l \end{array}$$

So P and F are *past* and *future* modalities.

Let us see how this idea fares with some examples. We should think of an atomic proposition as representing an untensed assertion. After a moment's thought, one can

The oracle speaks	p	The oracle spoke/has spoken	Pp
The oracle will speak	Fp	The oracle had spoken	PPp
The oracle will have spoken	FPp	The oracle never spoke	$\neg Pp$
There will be a time after which the oracle will not speak			$F\neg Fp$
There was a first time the oracle spoke			$P(p \wedge \neg Pp)$

Figure 1. Sentences and Priorean Translations

see that the very question of whether "untensed assertions" are possible will be a source of debate in this area. But let us ignore this and think of stative present tense assertions like *the oracle speaks* as an untensed assertion. Suppose that we take its semantics to be an atomic proposition p of the logic above. Then we can translate some English sentences as in Figure 1.

It is important to make a few comments about the contents of the figure. As with all translations from natural language into a formal language, one has to be clear on what has been achieved and what some of the problems are.

We also mention some ways that the system can be fruitfully extended. For example, it is straightforward to add binary modalities S and U for *since* and *until*. With a little more work, we can also add *now*. The simplest way to do this is to work on models $(L, <)$ with a distinguished l^* for the "present moment". Then we add to the clauses in (7) the following

$$l \models N\phi \quad \text{iff} \quad l^* \models \phi.$$

This proposal is due to Kamp [42], and it is discussed further in Burgess [11], Section 4B. Among the facts shown in these references is the fact that N is actually eliminable in this language. However, if one moves from a purely propositional setting one with more linguistically interesting phenomena, this reduction is rightfully lost. For example, consider

(8) i. The oracle predicted that there will be an earthquake.

ii. The oracle predicted that there would be an earthquake.

A natural representation of (1) is $Pr(o, NFe)$; the important point is that the future operator F is evaluated from the vantage point of "now". This contrasts with (2). Here a representation might be $Pr(o, Fe)$. The difference is that the prediction in (2) is that there will be an earthquake at some point later than the prediction, not the moment of utterance of the sentence.

There are some linguistic problems with any treatment of time as an extra parameter. One problem again concerns embedded tenses; these are especially interesting for modal logic since all of the important problems in modal logic arise precisely because modalities in formal systems may be iterated, and because accessibility relations in models can be deep. The natural symbolization of (6) in a modal approach comes out as something like

$$\begin{aligned} P \quad & ((\exists x)(\exists y)(\text{kindergarten}(x) \wedge \text{go}(J, x) \wedge \text{go}(y, x) \wedge \\ & F((\exists z)(\text{bank}(z) \wedge \text{president}(y, z))))). \end{aligned}$$

But then consider *John went to kindergarten with someone who has become a bank president.* Here the intended reading is that the person became a bank president before the utterance time. So having F in the scope of P in the representation would be a mistake. Another problem is that many temporal phenomena pertain more to events that distributed in time and hence do not admit a nice formulation: *Boris took piano lessons for six months.*

Sentences like Pp may be rendered either in the *simple past* as *The oracle spoke* or in the *present perfect* as *The oracle has spoken.* This means that whatever differences we ascribe to the two English forms will not be representable in the Priorean formalism.

Further, the logic contains forms like $PFPPFp$ which cannot be rendered into English except by transcribing the formal semantics into mathematical English. This is a problem not just for this work, but also for practically all accounts of any phenomenon which use recursion: the formalism will quickly contain forms not naturally renderable without heavy uses of devices like numbered or named pronouns.

For other natural English sentences that cannot be translated adequately in the Priorean formalism, consider *The oracle did not speak.* What we have here is an implicit reference to a particular time or set of times. So our sentence is not captured by $\neg Pp$, since that sentence amounts to a universal quantification over past times.

Furthermore, one would suspect that since we can add operators corresponding to *Since* and *Until*, we might also add an operator Y for *Yesterday.* Suppose our semantics makes use of a function $l \mapsto l-1$ and works by $l \models Y\phi$ iff $l-1 \models \phi$. However, here the a sentence like *Yesterday the oracle spoke* would correspond to Yp rather than YPp. So we are left with a puzzle about why the natural language sentence uses the past tense marker in the first place.

Hinrichs [37] noted that the sentence *Vincent left yesterday* has two natural renderings:

$$Y(P(leave(Vincent))) \qquad \text{and} \qquad P(Y(leave(Vincent))).$$

However, these both fail to have the intended meaning: the Y operator shifts the evaluation point to the previous day, but then the P operator takes the past from this point. A similar problem, noted by Partee [73] and reiterated in Hinrichs' paper is that tense and negation do not work well in Prior's approach. Translating "Vincent did not leave" by either $P(\neg leave(V))$ or the other alternative do not work.

Even though the rest of our discussion has dwelled the shortcomings of the Priorean approach, some aspects of the temporal system of language clearly are captured in it. Further discussion of tense logic and standard logic may be found in van Benthem [103, 104] and also Chapter 11 of this handbook. We also discuss an extension of Prior's approach due to Patrick Blackburn in Section 2.8 below.

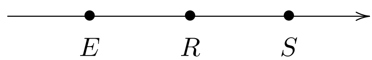
2.7 The reference time

One of the key contributions to this area comes in Reichenbach's textbook (on logic (!)) [82]. He points out that linguistic tense does not involve only "now" and "then" but also a third time, the *reference time.* So he described the tense system in terms of three times, S (the speech point), E (the event point), and R (the reference point). R is the time in a sentence "that we are really talking about".

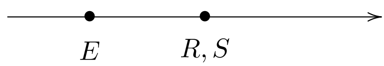
For example, consider the difference between the *simple past* and *past perfect* in English. The simple past is exemplified by *I saw John*; the past perfect by *I had seen John.*

The difference between these is that in the past perfect, E is prior to R (and both come before S): the speaker is describing an event from a vantage point (R) which is later than the time (E) of the event itself. In contrast, in the simple past, the event time and reference time are the same.

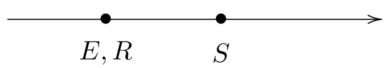
Incidentally, although Reichenbach seems to have preferred to think in terms of R, E, and S as *points*, there is good reason to prefer to take them to be *intervals*. The use of intervals in tense logic is a natural move, and many semantics papers do in fact make it. Here are some examples of the way various tense and aspect combinations in Reichenbach's system come out when we take R, E, and S to be intervals, writing $<$ for the subinterval relation:



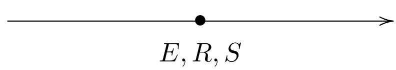
the past perfect: I had seen John.



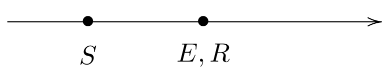
the present perfect: John has been to Boston.



the simple past: I gave a lecture.



the present: I name this ship *Chips*.



the simple future: I lecture (next week).

We have completed a quick tour of some of the central proposals concerning temporal ontology. It goes without saying that we have barely scratched the surface, that further work is in large measure concerned with corrections and criticisms of the classical ideas, etc. But we would be remiss in ending without mentioning that much of the current work is concerned not with points of time but rather with *events*; see for example, the book *The Proper Treatment of Events* [105]. Figure 2 contains a chart of some of the Reichenbach examples, worked in terms of events for S, E and R. The relation $<$ is that of *subevent*.

2.8 Temporal reference and hybrid logic

Reference to specific times can be incorporated into a Prior-style formalism by using ideas from hybrid logic (see Chapter 14 of this handbook). The basic idea of hybrid logic is to add a new sort of propositional symbol to the underlying modal language; these symbols are called *nominals*, and they are typically written i, j, and k. When working with nominals, one adds a semantic constraint that they be true at exactly one point. In this way, nominals 'name' the unique point they are true at.

This gives us a way of coping with some of the difficulties noted earlier. For example, we saw that *The oracle could not speak* could not be translated into the standard Priorean formalism; the simple representation $\neg Pp$ amounts to universal quantification over past times. But with the aid of nominals we have a better representation:

$$P(i \wedge \textit{the oracle not speak}).$$

This anchors the silence of the oracle at a particular time in the past, namely the time named by the nominal i.

Moreover, we now have a way handling reference times. Consider the sentence *The oracle had not spoken.* This picks out some past time (the reference time) and locates the silence of the oracle before that. This cannot be handled in the standard Priorean formalism, but once again with the aid of nominals, we can capture its meaning:

$$P(i \wedge P(\textit{the oracle not speak})).$$

This formula says that there is some time in the past (namely the one named by i) and that *before that* the oracle did not speak. In fact, as Blackburn [6] observes, all of Reichenbach's analyses can be handled in this way; the required hybrid representations are given in Figure 3.

It's also worth remarking that the ideas of hybrid logic combine naturally with multi-dimensional modal semantics of the type mentioned above. For example, Blackburn [6] uses this style of semantics to interpret propositional symbols like *yesterday*. The use of such symbols avoids the problems associated with the yesterday operator Y. For example, the hybrid representation of *Vincent left yesterday* would be

$$P(\textit{yesterday} \wedge \textit{Vincent leave}),$$

and this has the required interpretation. Moreover, the hybrid approach also correctly classifies sentences such as *Vincent will leave yesterday* as semantically anomalous. This sentence would be represented by the hybrid formula

$$F(\textit{yesterday} \wedge \textit{Vincent leave}).$$

This formula is false at all points in all models, hence the anomaly.

	Past	Present	Future
Simple	$E = R$, $R < S$ Mary saw John	$E = R = S$ Mary sees John	$E = R$, $S < R$ Mary will see John
Perfect	$E < R < S$ Mary had seen John	$E < R = S$ Mary has seen John	$E < R$, $S < R$ Mary will have seen John
Progressive	$E = R$, $R < S$ Mary was seing John	$E = R = S$ Mary is seeing John	$E = R$, $S < R$ Mary will be seeing John

Figure 2. Tense and Aspect in Terms of E, R, and S

Structure	**Name**	**English example**	**Representation**
E–R–S	Pluperfect	I had seen	$P(i \wedge P\varphi)$
E,R–S	Past	I saw	$P(i \wedge \varphi)$
R–E–S	Future-in-the-past	I would see	$P(i \wedge F\varphi)$
R–S,E	Future-in-the-past	I would see	$P(i \wedge F\varphi)$
R–S–E	Future-in-the-past	I would see	$P(i \wedge F\varphi)$
E–S,R	Perfect	I have seen	$i \wedge P\varphi$
S,R,E	Present	I see	$i \wedge \varphi$
S,R–E	Prospective	I am going to see	$i \wedge F\varphi$
S–E–R	Future perfect	I will have seen	$F(i \wedge P\varphi)$
S,E–R	Future perfect	I will have seen	$F(i \wedge P\varphi)$
E–S–R	Future perfect	I will have seen	$F(i \wedge P\varphi)$
S–R,E	Future	I will see	$F(i \wedge \varphi)$
S–R–E	Future-in-the-future	(Latin: abiturus ero)	$F(i \wedge F\varphi)$

Figure 3. Reichenbach's analysis in hybrid logic

2.9 A note on multidimensionality

One very interesting application of modal logic to semantic analysis is the use of multidimensional modal logic in connection with cross-world comparatives. Consider, for example, sentences like *This article is shorter than it might have been.* One approach to its semantics is to use not just a single world in the semantics, but to move to two or even more worlds. We might have an "actual" world and an "evaluation" world. For applications in semantics related to comparatives, see Cantwell [13]. By now there are also quite sophisticated modelings of tense and aspect; see, for example ter Meulen [94].

2.10 Problems and prospects

Even with the move to a Reichenbachian treatment of tense and aspect, there are remaining stubborn problems. Many of these are especially pertinent to the discussion of the application of possible world semantics; they indeed cause one to either re-think the use of possible worlds, or to propose modifications or extensions of it. Consider, for example, the *present relevance* of the perfect. For example, *Jimmy has lost his mind* intuitively entails that Jimmy has not gotten it back.

It is also important to note that a lot of real-world knowledge goes into judgments about sentences using time and tense. For example

(9) i. ??James McCawley has written many books.

ii. James McCawley wrote many books.

The first is anomalous, but only to one who knows that McCawley died in 1999. For someone who didn't know this, (9i) carries the implicature that McCawley is still alive. The point here is that (9i) and (9ii) are not equivalent, but the difference is due to background knowledge. So the entire system of time/tense/aspect interacts with the knowledge background of speakers and hearers. With this in mind, consider also, and note the difference between

Category	Description	Examples
S	Sentence	*John seek a unicorn*
CN	Common nouns	*man, woman, unicorn*
IV	Intransitive verb phrases	*walk*
S/IV	Noun phrases	*the man, every unicorn, John*
(S/IV)/CN	Determiners	*every, a, one, the*
IV/(S/IV)	Transitive verb phrases	*love, seek*
IV/S	Sentential complement verbs	*believe, hope, doubt*
CN/CN	Adjectives	*red, fake*
S/S	Sentential adverbs	*frequently, necessarily*

Figure 4. Categories and Sample Expressions in a Categorial Grammar

(10) i. The authors have regretted that they never met McCawley.

ii. The authors had regretted that they never met McCawley.

Many recent papers and books in the area emphasize the presence of causality and real-world knowledge in discourse about time; see Steedman [91]. Another book on this topic which emphasizes connections to logic programming and even robotics is van Lambalgen and Hamm [105].

2.11 Montague semantics

As was mentioned above, Montague's application of higher-order intensional logic marked the starting point for applications of modal logic in natural language semantics. Montague developed his theory of natural language semantics over the course of three papers (collected in [99]), each of which differs from the others in some respect. In the following, we give a survey of a "streamlined" approach, taken from Gamut [32].

Montague semantics consists of three parts: syntactic categories, semantic types, and operations on the members of each of these, where each operation on members of syntactic categories has a corresponding operation on the members of the corresponding semantic types. This correspondence is Montague's formalization of the principle of compositionality.

The theory of syntactic categories assumed in Montague semantics is loosely based on categorial grammar. The categories of categorial grammar are either basic categories or derived categories, which are formed by closing the basic categories under two operators: / and \. An expression that has a category of the form A/B is "looking to its right for" an expression of the category B to make an expression of the category A. And an expression of category $B\backslash A$ is looking to its left for an expression of category B to again make one of category A. Thus, derived categories have a functional behavior.

The full set of categories, CAT, is obtained by closing the basic categories, S (for *sentence*), CN (*common noun*), IV (*intransitive verb*), under the / operator. Thus, only one of the two operators of categorial grammar is used by Montague. Figure 4 has examples of the most common categories and some of their expressions.

At this point, we have described the set of syntactic categories and given examples. One forms the full set of expressions of the various categories by juxtaposition following

the categorial rules. For example, *John walks* is an S because *John* is $\mathsf{S/IV}$ and *walks* is an IV. From this, *believes John walks* is an IV. And then *Mary believes John walks* is again an S. This is as it should be, since we have a grammatical sentence.

We return to the syntactic categories and expressions below, after a digression concerning the formal semantics. Let e and t be any distinct objects, and define the set T of *semantic types* by the following inductive definition:

1. $e, t \in T$,
2. if $a, b \in T$, then $\langle a, b\rangle \in T$,
3. if $a \in T$, then $\langle s, a\rangle \in T$.

The idea is that e stands for *entity* and t for *truth value*, s for a set of possible worlds, and $\langle a, b\rangle$ for the set of all functions from a to b (or rather for the set of functions from the set that a stands for to the set b stands for). The difference between syntactic categories and semantic types is that syntactic categories have a notion of "order" built-in. In the following, we will use upper-case letters for syntactic categories and lower-case letters for semantic types.

Given infinite sets of variables for each type a, denoted by VAR_a, and, possibly empty, sets of constants for each type a, denoted by CON_a, we define the well-formed expressions of type a, denoted by WE_a, as follows:

1. $\mathsf{VAR}_a \subseteq \mathsf{WE}_a$ and $\mathsf{CON}_a \subseteq \mathsf{WE}_a$,
2. if $\alpha \in \mathsf{WE}_{\langle a,b\rangle}$ and $\beta \in \mathsf{WE}_a$, then $\alpha(\beta) \in \mathsf{WE}_b$,
3. if $\varphi, \psi \in \mathsf{WE}_t$, then $\neg\varphi \in \mathsf{WE}_t$ and $(\varphi \wedge \psi) \in \mathsf{WE}_t$,
4. if $\varphi \in \mathsf{WE}_t$ and $v \in \mathsf{VAR}_a$, then $\forall v\varphi \in \mathsf{WE}_t$,
5. if $\alpha, \beta \in \mathsf{WE}_a$, then $\alpha = \beta \in \mathsf{WE}_t$,
6. if $\alpha \in \mathsf{WE}_a$ and $v \in \mathsf{VAR}_b$, then $\lambda v\alpha \in \mathsf{WE}_{\langle b,a\rangle}$,
7. if $\varphi \in \mathsf{WE}_t$, then $\Box\varphi \in \mathsf{WE}_t$,
8. if $\alpha \in \mathsf{WE}_a$, then ${}^\wedge\alpha \in \mathsf{WE}_{\langle s,a\rangle}$,
9. if $\alpha \in \mathsf{WE}_{\langle s,a\rangle}$, then ${}^\vee\alpha \in \mathsf{WE}_a$.

We will use other connectives: $\Diamond, \vee, \rightarrow, \leftrightarrow, \exists$, to abbreviate their usual definitions in terms of the connectives above. The reason that these types are referred to as *semantic* types is that each type has a corresponding domain. Given a set of individuals, D, and a set of worlds, W, we define the domain of a type t, denoted by $\mathcal{D}_{t,D,W}$ as follows:

1. $\mathcal{D}_{e,D,W} = D$
2. $\mathcal{D}_{t,D,W} = \{0, 1\}$
3. $\mathcal{D}_{\langle a,b\rangle,D,W} = \mathcal{D}_{b,D,W}^{\mathcal{D}_{a,D,W}}$
4. $\mathcal{D}_{\langle s,a\rangle,D,W} = \mathcal{D}_{a,D,W}^{W}$

where A^B denotes the set of functions from B to A. We now define the interpretation of expressions. A *model*, $\mathcal{M}$, is a triple, (D, W, I), where D is a non-empty set of individuals, W is a non-empty set of worlds, and I is an interpretation of the constants at a world. We define $[\![\alpha]\!]_{\mathcal{M},w,g}$, where $\mathcal{M}$ is a model, $w \in W$ is a world, and g is a variable assignment. As usual, we denote the variable assignment that differs from g at most in that it assigns d to v by $g[v/d]$.

1. if $\alpha \in \mathsf{CON}_a$, then $[\![\alpha]\!]_{\mathcal{M},w,g} = I(w, \alpha)$; if $\alpha \in \mathsf{VAR}_a$, then $[\![\alpha]\!]_{\mathcal{M},w,g} = g(\alpha)$,
2. if $\alpha \in \mathsf{WE}_{\langle a,b\rangle}$ and $\beta \in \mathsf{WE}_a$, then $[\![\alpha(\beta)]\!]_{\mathcal{M},w,g} = [\![\alpha]\!]_{\mathcal{M},w,g}([\![\beta]\!]_{\mathcal{M},w,g})$,
3. if $\varphi, \psi \in \mathsf{WE}_t$, then $[\![\neg\varphi]\!]_{\mathcal{M},w,g} = 1$ iff $[\![\varphi]\!]_{\mathcal{M},w,g} = 0$, and $[\![(\varphi \wedge \psi)]\!]_{\mathcal{M},w,g} = 1$ iff $[\![\varphi]\!]_{\mathcal{M},w,g} = 1$ and $[\![\psi]\!]_{\mathcal{M},w,g} = 1$,
4. if $\varphi \in \mathsf{WE}_t$ and $v \in \mathsf{VAR}_a$, then $[\![\forall v\varphi]\!]_{\mathcal{M},w,g} = 1$ iff for all $d \in \mathcal{D}_a$, $[\![\varphi]\!]_{\mathcal{M},w,g[v/d]} = 1$,
5. if $\alpha, \beta \in \mathsf{WE}_a$, then $[\![\alpha = \beta]\!]_{\mathcal{M},w,g} = 1$ iff $[\![\alpha]\!]_{\mathcal{M},w,g} = [\![\beta]\!]_{\mathcal{M},w,g}$,
6. if $\alpha \in \mathsf{WE}_a$ and $v \in \mathsf{VAR}_b$, then $[\![\lambda v\alpha]\!]_{\mathcal{M},w,g}$ is the function $h \in \mathcal{D}_{\langle b,a\rangle}$, such that for all $d \in \mathcal{D}_b$, $h(d) = [\![\alpha]\!]_{\mathcal{M},w,g[v/d]}$,
7. if $\varphi \in \mathsf{WE}_t$, then $[\![\Box\varphi]\!]_{\mathcal{M},w,g} = 1$ iff for all $w' \in W$, $[\![\varphi]\!]_{\mathcal{M},w',g} = 1$,
8. if $\alpha \in \mathsf{WE}_a$, then $[\![{}^\wedge\alpha]\!]_{\mathcal{M},w,g}$ is the function $h \in \mathcal{D}_{\langle s,a\rangle}$, such that for all $w' \in W, h(w') = [\![\alpha]\!]_{\mathcal{M},w',g}$,
9. if $\alpha \in \mathsf{WE}_{\langle s,a\rangle}$, then $[\![{}^\vee\alpha]\!]_{\mathcal{M},w,g} = [\![\alpha]\!]_{\mathcal{M},w,g}(w)$.

This allows us to have formal terms for what we informally wrote above in (3).

There is a symmetry between λ-abstraction and application (i.e. β conversion), and $^\wedge$ abstraction and $^\vee$ application. However, while the following form of β-conversion can only be applied in the *extensional* fragment of this system, it only holds in restricted cases in the intensional system.

THEOREM 1. *In the extensional fragment of $\lambda x\beta(\gamma)$ is equivalent to $\beta[x \mapsto \gamma]$ if all free variables in γ are free for x in γ.*

However, in the extensional system this equivalence fails. It is possible to extend this equivalence to a restricted set of expressions of the intensional system: the intensionally closed expressions, whose extension does not vary from world to world. The *intensionally closed expressions* in L, denoted by ICE^L, is the minimal subset of WE^L such that

1. If $x \in \mathsf{VAR}_a$, then $v \in \mathsf{ICE}^L$,
2. If $\alpha \in \mathsf{WE}^L_a$, then ${}^\wedge\alpha \in \mathsf{ICE}^L$
3. If $\varphi \in \mathsf{WE}^L_t$, then $\Box\varphi \in \mathsf{ICE}^L$
4. ICE^L is closed under boolean connectives, quantifiers, and λ-abstraction.

The above mentioned symmetry is summarized in the following two theorems:

THEOREM 2. ${}^\vee({}^\wedge\alpha)$ *is equivalent to* α.

THEOREM 3. $\lambda x\beta(\gamma)$ *is equivalent to* $\beta[x \mapsto \gamma]$ *if*

1. *all free variables in* γ *are free for* x *in* γ*; and*
2. *either* $\gamma \in \mathsf{ICE}^L$*, or no free occurrence of* x *in* β *lies within the scope of* $\Box, {}^\wedge$*.*

Now, we associate semantic types with syntactic categories as follows using the following function f:

$$f(\mathsf{S}) = t$$
$$f(\mathsf{CN}) = f(\mathsf{IV}) = \langle e, t\rangle$$
$$f(\mathsf{A/B}) = \langle\langle s, f(\mathsf{B})\rangle, f(\mathsf{A})\rangle$$

With each syntactic category, A, we associate a set of *basic expression* of that category, denoted by B_{A}, and a set of expressions of that category, denoted by P_{A}.

The next step involves the definition of syntactic operations that create complex expressions. In the following, we will use the same rule numbers as [32]. Here are the first three rules:

$$B_{\mathsf{A}} \subseteq P_{\mathsf{A}} \tag{S1}$$

If $\alpha \in P_{\mathsf{S/IV}}$ and $\beta \in P_{\mathsf{IV}}$, then $F_1(\alpha,\beta) \in P_{\mathsf{S}}$, and $F_1(\alpha,\beta) = \alpha\beta'$, where β' (S2)
is the result of replacing the main verb in β by its third-person singular present form.

If $\alpha \in P_{(\mathsf{S/IV})/\mathsf{CN}}$ and $\beta \in P_{\mathsf{CN}}$, then $F_2(\alpha,\beta) \in P_{\mathsf{S/IV}}$, and $F_2(\alpha,\beta) = \alpha\beta$. (S3)

Rule S1 simply makes the basic expressions of category A expressions of category A. Rule S2 combines noun phrases with verb phrases to make sentences, the side condition enforcing subject-verb agreement. Rule S3 combines determiners with common nouns to form noun phrases.

EXAMPLE 4. Here is a derivation for "every man walks":

$$F_2(\textit{every}, \textit{man}) = \textit{every man}$$
$$F_1(\textit{every man}, \textit{walk}) = \textit{every man walks}$$

Since the syntactic derivations in Montague grammar are very straightforward, we will dispense with them in the rest of this article.

We now define a function $\mapsto$ that associates with each expression of category A an expression of type $f(\mathsf{A})$, its *meaning*. First, the translation of most basic expressions will simply be a constant. We will denote the constants corresponding to basic expressions using CAPS. Thus, the constant corresponding to *walk* is WALK. The only exceptions to this rule are noun phrases, determiners, the verb *be*, and *necessarily*.

$$T1:$$
$$John \mapsto \lambda X({}^\vee X(j))$$
$$Mary \mapsto \lambda X({}^\vee X(m))$$
$$he_n \mapsto \lambda X({}^\vee X(x_n))$$
$$every \mapsto \lambda Y \lambda X \forall x({}^\vee Y(x) \rightarrow {}^\vee X(x))$$

$$\text{If } \alpha \in P_{\mathsf{S/IV}}, \beta \in P_{\mathsf{IV}}, \alpha \mapsto \alpha', \text{ and } \beta \mapsto \beta', \text{ then } F_1(\alpha,\beta) \mapsto \alpha'({}^\wedge\beta') \tag{T2}$$
$$\text{If } \alpha \in P_{(\mathsf{S/IV})/\mathsf{CN}}, \beta \in P_{\mathsf{CN}}, \alpha \mapsto \alpha', \text{ and } \beta \mapsto \beta', \text{ then } F_2(\alpha,\beta) \mapsto \alpha'({}^\wedge\beta') \tag{T3}$$

EXAMPLE 5. Here is the translation for "every man walks."

$$\cfrac{\cfrac{\cfrac{\cfrac{\lambda Y \lambda X \forall x({}^{\vee}Y(x) \rightarrow^{\vee} X(x)) \quad \textsc{man}}{\lambda X \forall x(\textsc{man}(x) \rightarrow^{\vee} X(x))}\ T2 \qquad \textsc{walk}}{\lambda X \forall x(\textsc{man}(x) \rightarrow^{\vee} X(x))({}^{\wedge}\textsc{walk})}\ T3}{\forall x(\textsc{man}(x) \rightarrow^{\vee} ({}^{\wedge}\textsc{walk}(x)))}\ \beta\text{-conversion}}{\forall x(\textsc{man}(x) \rightarrow \textsc{walk}(x))}\ {}^{\vee\wedge}\text{-cancellation}$$

Now we will consider transitive verbs. Again, we first give a syntactic rule, followed by a semantic rule:

If $\alpha \in P_{\mathsf{IV/(S/IV)}}$ and $\beta \in P_{\mathsf{S/IV}}$, then $F_6(\alpha, \beta) \in P_{\mathsf{IV}}$, and $F_6(\alpha, \beta) = \alpha\beta'$ (S7)
where $\beta'$is the accusative form of β if β is a syntactic variable; otherwise $\beta' = \beta$.
If $\alpha \in P_{\mathsf{S/IV}}, \beta \in P_{\mathsf{IV}}, \alpha \mapsto \alpha'$, and $\beta \mapsto \beta'$, then $F_6(\alpha, \beta) \mapsto \alpha'({}^{\wedge}\beta')$. (T7)

We will use the following two notational conventions from Gamut [32]. The first is just an instance of what computer scientists call *uncurrying*.

If γ is an expression of type $\langle a, \langle b, t\rangle\rangle, \alpha$ and expression of type a, and β (NC1)
an expression of type b, then we may write $\gamma(\beta, \alpha)$ instead of $(\gamma(\alpha))(\beta)$.

Before discussing the second notational convention ($NC2$), we need to review how Montague proposed to treat transitive verbs. In Montague's system, the meanings transitive verbs are relations between individuals and second-order properties, i.e. they are of type $\langle\langle s, \langle\langle s, \langle e, t\rangle\rangle, t\rangle\rangle, \langle e, t\rangle\rangle$. Thus, since they are not relations between individuals, we can have a statement such as "John seeks a unicorn" be true, without it entailing the existence of unicorns. However, there are certain transitive verbs, so-called *extensional transitive verbs* which entail the existence of their arguments. For these expressions NC2 will allow us to move from Montague's higher-order interpretation of transitive verbs to relations between individuals.

If δ is an expression of type $\langle\langle s, \langle\langle s, \langle e, t\rangle\rangle, t\rangle\rangle, \langle e, t\rangle\rangle$, then we may write δ_* (NC2)
instead of $\lambda y \lambda x \delta(x, {}^{\wedge} \lambda X {}^{\vee} X(y))$.

The expression δ_* refers to the relation that holds between x and y iff the relation δ holds between x and the intension of the set of all properties of y, i.e. ${}^{\wedge}\lambda X {}^{\vee} X(y)$. For further details and discussion, see Gamut [32].

EXAMPLE 6. Here is a derivation for "every man loves a woman."

$$\lambda Y\lambda X\exists x({}^{\vee}Y(x) \to^{\vee} X(x)) \quad \textsc{woman}$$
$$\vdots$$
$$\lambda Y\lambda X\forall y({}^{\vee}Y(y) \to^{\vee} X(y)) \quad \textsc{man} \qquad \dfrac{\textsc{love} \qquad \lambda X\exists x(\textsc{woman}(x) \wedge^{\vee} X(x))}{\textsc{love}({}^{\wedge}\lambda X\exists x(\textsc{woman}(x) \wedge^{\vee} X(x)))}\,T7$$
$$\vdots$$
$$\dfrac{\lambda Y\forall y(\textsc{man}(y) \to^{\vee} Y(y)) \qquad \textsc{love}({}^{\wedge}\lambda X\exists x(\textsc{woman}(x) \wedge^{\vee} X(x)))}{\lambda Y\forall y(\textsc{man}(y) \to^{\vee} Y(y))({}^{\wedge}\textsc{love}({}^{\wedge}\lambda X\exists x(\textsc{woman}(x) \wedge^{\vee} X(x))))}\,T2$$
$$\dfrac{\lambda Y\forall y(\textsc{man}(y) \to^{\vee} Y(y))({}^{\wedge}\textsc{love}({}^{\wedge}\lambda X\exists x(\textsc{woman}(x) \wedge^{\vee} X(x))))}{\forall y(\textsc{man}(y) \to^{\vee} ({}^{\wedge}\textsc{love}({}^{\wedge}\lambda X\exists x(\textsc{woman}(x) \wedge^{\vee} X(x))))(y))}\,\beta\text{-conversion}$$
$$\dfrac{\forall y(\textsc{man}(y) \to^{\vee} ({}^{\wedge}\textsc{love}({}^{\wedge}\lambda X\exists x(\textsc{woman}(x) \wedge^{\vee} X(x))))(y))}{\forall y(\textsc{man}(y) \to (\textsc{love}({}^{\wedge}\lambda X\exists x(\textsc{woman}(x) \wedge^{\vee} X(x))))(y))}\,{}^{\vee\wedge}\text{-cancellation}$$
$$\dfrac{\forall y(\textsc{man}(y) \to (\textsc{love}({}^{\wedge}\lambda X\exists x(\textsc{woman}(x) \wedge^{\vee} X(x))))(y))}{\forall y(\textsc{man}(y) \to \textsc{love}(y,{}^{\wedge}\lambda X\exists x(\textsc{woman}(x) \wedge^{\vee} X(x)))}\,NC1$$
$$\dfrac{\forall y(\textsc{man}(y) \to \textsc{love}(y,{}^{\wedge}\lambda X\exists x(\textsc{woman}(x) \wedge^{\vee} X(x)))}{\forall y(\textsc{man}(y) \to \exists x(\textsc{woman}(x) \wedge \textsc{love}_*(y,x)))}\,NC2$$

The following rule schema is used to create different derivations for ambiguous sentences.

If $\alpha \in P_{\mathsf{S/IV}}$ and $\varphi \in P_{\mathsf{S}}$, then $F_{7,n}(\alpha,\varphi) = \varphi'$, where φ' is the result of the following substitution in φ: (S8,n)
(i) if α is not a syntactic variable he_k, then replace the first occurrence of he_n or him_n with α, and the other occurrences of he_n or him_n with the appropriate anaphoric pronoun.;
(ii) if $\alpha = he_k$, then replace every occurrence of he_n and of him_n with him_k.

And the corresponding semantic rule:

$$\text{If } \alpha \in P_{\mathsf{S/IV}}, \varphi \in P_{\mathsf{S}}, \alpha \mapsto \alpha', \text{ and } \varphi \mapsto \varphi', \text{ then } F_{7,n}(\alpha,\varphi) \mapsto \alpha'({}^{\wedge}\lambda x_n\varphi') \qquad \text{(T8,n)}$$

EXAMPLE 7. Here is another derivation of "every man loves a woman." This time, "everybody loves him$_1$" is derived first, and "a woman" is *quantified in.*

$$\lambda Y\lambda X\exists x({}^{\vee}Y(x) \to^{\vee} X(x)) \quad \textsc{woman} \qquad \textsc{love} \quad \lambda X^{\vee}X(x_1)$$
$$\vdots \qquad\qquad\qquad \vdots$$
$$\dfrac{\lambda X\exists x(\textsc{woman}(x) \wedge^{\vee} X(x)) \qquad \forall y(\textsc{man}(y) \to \textsc{love}_*(y,x_1))}{\lambda X\exists x(\textsc{woman}(x) \wedge^{\vee} X(x))({}^{\wedge}\lambda x_1\forall y(\textsc{man}(y) \to \textsc{love}_*(y,x_1)))}\,T8,1$$
$$\dfrac{\lambda X\exists x(\textsc{woman}(x) \wedge^{\vee} X(x))({}^{\wedge}\lambda x_1\forall y(\textsc{man}(y) \to \textsc{love}_*(y,x_1)))}{\exists x(\textsc{woman}(x) \wedge^{\vee} ({}^{\wedge}\lambda x_1\forall y(\textsc{man}(y) \to \textsc{love}_*(y,x_1))(x)))}\,\beta\text{-conversion}$$
$$\dfrac{\exists x(\textsc{woman}(x) \wedge^{\vee} ({}^{\wedge}\lambda x_1\forall y(\textsc{man}(y) \to \textsc{love}_*(y,x_1))(x)))}{\exists x(\textsc{woman}(x) \wedge (\lambda x_1\forall y(\textsc{man}(y) \to \textsc{love}_*(y,x_1))(x)))}\,{}^{\vee\wedge}\text{-cancellation}$$
$$\dfrac{\exists x(\textsc{woman}(x) \wedge (\lambda x_1\forall y(\textsc{man}(y) \to \textsc{love}_*(y,x_1))(x)))}{\exists x(\textsc{woman}(x) \wedge (\forall y(\textsc{man}(y) \to \textsc{love}_*(y,x))))}\,\beta\text{-conversion}$$

Now we can use quantifying-in to derive multiple interpretations of *believe*-sentences, corresponding to the above-mentioned *de dicto-de re* distinction. Assuming that the syntactic category of *believe* is $\mathsf{IV/S}$, we need new syntactic and semantic rules for such expressions:

$$\text{If } \alpha \in P_{\mathsf{IV/S}} \text{ and } \beta \in P_{\mathsf{S}}, \text{ then } F_{11}(\alpha,\beta) \in P_{\mathsf{IV}}, \text{ and } F_{11}(\alpha,\beta) = \alpha\beta \qquad \text{(S15)}$$
$$\text{If } \alpha \in P_{\mathsf{IV/S}}, \varphi \in P_{\mathsf{S}}, \alpha \mapsto \alpha', \text{ and } \varphi \mapsto \varphi', \text{ then } F_{11}(\alpha,\varphi) \mapsto \alpha'({}^{\wedge}\varphi') \qquad \text{(T15)}$$

EXAMPLE 8. Here are derivations for *John believes that a man walks.* The first gives us the *de dicto* reading:

$$\lambda Y \lambda X \exists x({}^{\vee}Y(x) \wedge {}^{\vee} X(x)) \quad \text{MAN}$$
$$\vdots$$
$$\lambda X \exists x(\text{MAN}(x) \wedge {}^{\vee} X(x)) \qquad \text{WALK}$$
$$\vdots$$
$$\lambda X({}^{\vee}X(j)) \qquad \dfrac{\text{BELIEVE} \qquad \exists x(\text{MAN}(x) \wedge \text{WALK}(x))}{\text{BELIEVE}({}^{\wedge}\exists x(\text{MAN}(x) \wedge \text{WALK}(x))}\ T15$$
$$\vdots$$
$$\text{BELIEVE}(j, ({}^{\wedge}\exists x(\text{MAN}(x) \wedge \text{WALK}(x)))$$

While the quantifying-in version gives the *de re* reading:

$$\lambda Y \lambda X \exists x({}^{\vee}Y(x) \rightarrow {}^{\vee} X(x)) \quad \text{MAN} \qquad \lambda X {}^{\vee} X(x_1) \quad \text{WALK}$$
$$\vdots \qquad\qquad\qquad\qquad \vdots$$
$$\dfrac{\lambda X \exists x(\text{MAN}(x) \wedge {}^{\vee} X(x)) \qquad \text{BELIEVE}(j, ({}^{\wedge}\text{WALK}(x_1))}{\lambda X \exists x(\text{MAN}(x) \wedge {}^{\vee} X(x))({}^{\wedge}\lambda x_1 \text{BELIEVE}(j, ({}^{\wedge}\text{WALK}(x_1)))}\ T8, 1$$
$$\vdots$$
$$\exists x(\text{MAN}(x) \wedge \text{BELIEVE}(j, ({}^{\wedge}\text{WALK}(x))))$$

In addition to epistemic modalities, we can give a treatment of of sentential adverbs, like *necessarily*, using the following two rules:

$$\text{If } \alpha \in P_{\mathsf{S/S}} \text{ and } \beta \in P_{\mathsf{S}}, \text{ then } F_{11}(\alpha, \beta) \in P_{\mathsf{S}}, \text{ and } F_{11}(\alpha, \beta) = \alpha\beta \tag{S20}$$
$$\text{If } \alpha \in P_{\mathsf{S/S}}, \varphi \in P_{\mathsf{S}}, \alpha \mapsto \alpha', \text{ and } \varphi \mapsto \varphi', \text{ then } F_{11}(\alpha, \varphi) \mapsto \alpha'({}^{\wedge}\varphi') \tag{T20}$$

However, we need to distinguish non-logical adverbs, like *rarely*, from logical ones, like *necessarily*, because we want the meaning of the former, but not the latter, to vary across models. This is accomplished through *meaning postulates*, an invention of Carnap's [14], which can be used to relate logical constants with expressions and also to relate expressions with each other, e.g. $\mathsf{bachelor}$ with $\neg\mathsf{married}$. The relevant meaning postulate for *necessarily* would be:

$$\forall p \Box(\text{NECESSARILY}(p) \leftrightarrow \Box^{\vee} p)$$

We conclude this section with the example: *Necessarily, John walks.*

EXAMPLE 9.

$$\lambda X({}^{\vee}X(j)) \quad \text{WALK}$$
$$\vdots$$
$$\dfrac{\dfrac{\dfrac{\text{NECESSARILY} \qquad \text{WALK}(j)}{\text{NECESSARILY}({}^{\wedge}\text{WALK}(j))}\ T20}{\Box^{\vee}({}^{\wedge}\text{WALK}(j))}\ MP}{\Box(\text{WALK}(j))}\ {}^{\vee\wedge}\text{-cancellation}$$

The preceding examples are meant to be illustrative of the way that higher-order intensional logic is used in Montague grammar to model the meanings of natural language

expressions. There are many shortcomings of Montague's framework, some formal and some empirical, the rectification of which can be seen as the motivation for the majority of the current approaches in semantic theory. See [24] and [75] for additional details of Montague grammar and for discussions of its shortcomings.

3 SYNTAX

3.1 Mathematical linguistics

Mathematical linguistics is concerned with models of natural languages and linguistic theories and their formal properties, especially those theories about syntax. The properties of interest are typically those of theoretical computer science, particularly from formal language theory, complexity theory, and learnability theory. Applications of logic to syntax combine the first two by giving tools to assess the complexity of formal languages descriptively. This line of research is related to finite model theory [25] and descriptive complexity theory [40].

We will review the basics of formal language theory as it applies to this setting. For a more detailed introduction see Partee, ter Meulen, and Wall [72], or, at a more advanced level and including many applications of modal logic, Kracht [54]. In order to model natural language syntax mathematically, we use strings. An *alphabet* is a finite set Σ of *symbols*, and a *string* over Σ is a finite sequence of elements of Σ. This includes the empty sequence, denoted by ε. A fundamental operation on strings is *string concatenation* which we will denote by juxtaposition. We denote the set of all strings over Σ by Σ^*, and the set of all non-empty strings by Σ^+. A *(formal) language* is a set of strings; i.e., a subset of Σ^*. In the intended applications, Σ is the set of words (or even morphemes) of a natural language (rather than just letters of some alphabet), however, for the purpose of examples we will frequently just use letters. Mathematical linguistics uses formal languages as models of natural languages: we identify English with the set of English sentences.

The purpose of grammatical theories is to distinguish the well-formed (grammatical) strings from the ill-formed (ungrammatical) strings. This can be achieved in a number of ways, using automata, grammars, algebras, or logic. However, it is an important assumption of linguistic theory, dating back at least to American Structuralism, that sentences of natural languages are not just linear sequences, but that they contain hierarchical structures: constituents. Furthermore, the division of a string into constituents plays an important part in semantics, since the principle of compositionality stipulates that the meaning of a string depends on the meaning of the words and the way they are put together, the latter of which can be captured by the constituent structure. While automata, algebras, and logics can be used to define languages, formal grammars play a central role in mathematical linguistics because they can associate a hierarchical structure with the strings that they generate: the derivation tree. Note that not all formal grammars can associate derivation trees with the strings they generate, but for linguistic applications, grammars that do are typically more interesting. Thus, we distinguish the *weak generative capacity*, the set of strings generated by a grammar, from the *strong generative capacity*, the set of structural descriptions or trees assigned by the grammar to the strings that it generates.

We will be referring frequently to a particular class of grammars and the languages

they generate: the *context-free grammars* (CFGs) and *context-free languages* (CFLs). CFGs are specified in terms of two alphabets, Σ and Γ, which are called the *terminal* and *non-terminal* alphabets, respectively. The terminal alphabet consists of the symbols that make up the strings that the grammar generates; the non-terminal alphabet can be thought of as corresponding to the syntactic categories of traditional grammar. In addition, CFGs are specified in terms of *a finite set of rules*, P, and a distinguished member of Γ, *the start symbol*, denoted by S. The rules in P are of the form $A \to w$, where $w \in (\Sigma \cup \Gamma)^*$. CFGs derive strings of terminal symbols by successively rewriting non-terminal symbols. Let $x, y, z \in (\Sigma \cup \Gamma)^*$, and $A \in \Gamma$. We write $xAz \Rightarrow_G xyz$ to indicate that xyz can be obtained from xAz by using the rule $A \to y$ of G. We use $\Rightarrow^*_G$ to denote the reflexive, transitive closure of $\Rightarrow_G$. The language generated by a CFG, G, denoted by $L(G)$, is defined as

$$L(G) = \{w \mid w \in \Sigma^*, S \Rightarrow^*_G w\}$$

A language, L, is called a CFL if there is a CFG, G, such that $L = L(G)$.

The CFLs play a central role in mathematical linguistics; they are in some sense a yardstick, because they approximate many natural language languages reasonably well [33] and they can be processed efficiently [39]. On the other hand, many formalisms are defined for the explicit purpose of extending the weak generative capacity of CFGs, that is to obtain non-CF languages. The reason for this is that there are natural language phenomena that are not context-free [89]. There are also some proposals which only go beyond CFGs in terms of strong generative capacity, although they are weakly equivalent to CFGs. That is, they are interested in obtaining sets of structures which go beyond the sets of parse trees of CF languages, but which generate CF string languages. Both the Lambek calculus [101] and regular tree languages (see below) are examples of such proposals.

3.2 Preliminary: logics of strings

The logic of strings was first studied by logicians interested in decidability [10]. It was continued within formal language theory, and had an algebraic slant [77, 92]. The case of strings has not found many applications to linguistics, as the more involved settings below have. However, this simpler case is useful in getting an intuition about the more complex cases. For a more detailed introductions to this area see Khoussainov and Nerode [45] and Thomas [98].

First of all, our intended models are what we shall call *string structures*. These are Kripke frames of the form

$$1 \longrightarrow 2 \longrightarrow \quad \cdots \quad \longrightarrow n\ .$$

The idea is that a *string* on some alphabet gives rise to a frame as above. Adding a valuation amounts to specifying subsets of the model. Since we intend the atomic propositions to be the alphabet symbols, the subsets of the frame corresponding to these symbols correspond to the positions in the given word with the given symbols. Thus there is an extra condition that each world in the frame satisfy exactly one atomic sentence. A string model is a pair (W, v) consisting of a string structure W together with a valuation v that meets this extra condition. We usually omit the valuation from our notation. Let

Syntax	**Sentences** φ	$p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid [\rightarrow]\varphi \mid [\rightarrow^*]\varphi$
Semantics	**Main Clauses**	$W, i \models [\rightarrow]\varphi$ iff $W, i+1 \models \varphi$
		$W, i \models [\rightarrow^*]\varphi$ iff for all $j \geq i, W, j \models \varphi$

Figure 5. Modal logic of strings: $\mathcal{L}(\rightarrow, \rightarrow^*)$

Σ be an alphabet, considered also as a set of atomic propositions for our modal language. A set of string models over Σ corresponds to a subset of Σ^+ of non-empty words over Σ. The correspondence associates to the string model (W, v) the string $W_1 \cdots W_n$, where n is the length of W and each W_i is the unique element of Σ satisfied by i in the model W.

Modal Logic of Strings

Figure 5 contains the basic modal logic of strings which we will call $\mathcal{L}(\rightarrow, \rightarrow^*)$. The semantics in Figure 5 defines the relation $W, i \models \varphi$. We say that a string W satisfies a formula φ if $W, 1 \models \varphi$. A language L is *definable* in this (or another language) if there is a sentence φ so that L is exactly the set of strings satisfying a sentence φ.

In order to study the languages definable in $\mathcal{L}(\rightarrow, \rightarrow^*)$, we introduce a class of languages, called the *star-free* languages (the reason for the name will become apparent later). The star-free languages, which are defined in Figure 6, were introduced by McNaughton and Pappert [66] to study first-order (FO) definable languages. It should be noted that star-free languages are sets of strings, and as such might well contain the empty string ε. Since we are going to be interested in classes of models which correspond to strings, and since the carrier sets our models must be non-empty, we are going to be interested in *ε-free* star languages as defined in Figure 7.

Syntax	**Expressions** r	$0 \mid 1 \mid a \mid rs \mid r+s \mid -r$
Semantics		$[\![0]\!] = \varnothing$
		$[\![1]\!] = \{\varepsilon\}$
		$[\![a]\!] = \{a\}$
		$[\![rs]\!] = \{st \mid s \in [\![r]\!], t \in [\![s]\!]\}$
		$[\![r+s]\!] = [\![r]\!] \cup [\![s]\!]$
		$[\![-r]\!] = \Sigma^* - [\![r]\!]$

Figure 6. The syntax and semantics of star-free expressions

Syntax	**Expressions** r	$0 \mid a \mid rs \mid r+s \mid -r$
Semantics	**Main Clause**	$[\![-r]\!] = \Sigma^+ - [\![r]\!]$

Figure 7. The syntax and semantics of ε-free star-free expressions

Here are some examples of formulas in $\mathcal{L}(\rightarrow, \rightarrow^*)$ and the star-free languages that correspond to them (taken from [19]). Notice that we are using *regular expressions* (see below) to describe these star-free languages because they are shorter; these languages can also be described with star-free expressions.

Syntax	**Formulas** φ	$p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid [\rightarrow]\varphi \mid [\rightarrow^*]\varphi \mid \mathcal{U}(\varphi, \psi)$
Semantics	**Main Clause**	$W, i \models \mathcal{U}(\varphi, \psi)$ iff there exists a $j \geq i$, such that $W, j \models \varphi$, and for all $n, i \leq n \leq j, W, n \models \psi$

Figure 8. Temporal logic of strings: PTL

EXAMPLE 10. The language $(ab)^+$ is defined by the formula

$$a \wedge \langle\rightarrow^*\rangle(b \wedge \neg\langle\rightarrow\rangle a \wedge \neg\langle\rightarrow\rangle b) \wedge \neg\langle\rightarrow^*\rangle(a \wedge \langle\rightarrow\rangle a) \wedge \neg\langle\rightarrow^*\rangle(b \wedge \langle\rightarrow\rangle b)$$

Here and in the following, $\langle\cdot\rangle\varphi$ abbreviates $\neg[\cdot]\neg\varphi$. This formula says that the first letter is an a, the last letter is a b, and there are no consecutive a's or b's. Notice that $(aa)^+$ is not $\mathcal{L}(\rightarrow, \rightarrow^*)$ definable; in fact it is not even FO definable.

EXAMPLE 11. Let $A = \{a, b, c\}$. The language $A^*a(a+c)^*$ is defined by the formula

$$\langle\rightarrow^*\rangle(a \wedge \langle\rightarrow\rangle\neg\langle\rightarrow^*\rangle b)$$

The following proposition states the relationship between $\mathcal{L}(\rightarrow, \rightarrow^*)$ definability and star-freeness. There is also an algebraic characterization in Cohen *et al.* [19] which is omitted here.

PROPOSITION 12. *If $L \subseteq \Sigma^+$ is definable in $\mathcal{L}(\rightarrow, \rightarrow^*)$, then L is star-free.*

However, there are star-free languages that are not definable in $\mathcal{L}(\rightarrow, \rightarrow^*)$.

PROPOSITION 13. *The language $a^*b(a+b+c)^*$ is not $\mathcal{L}(\rightarrow, \rightarrow^*)$ definable.*

Proof. We use a version of Ehrenfeucht games for $\mathcal{L}(\rightarrow, \rightarrow^*)$ between word models W and V. The r-round game works exactly as in the standard game for modal logic on Kripke models. There are distinguished points in the two models, and they are updated in each round of a play. The difference from the standard games is that player I may decide at each round to play a standard move or else a $*$-*move*. In the $*$-move, I picks one of the two structures and moves the distinguished point (say w_i) to some w_j with $j \geq i$. Then II does the same in the other structure. If the distinguished points are labelled differently at any round, then I wins the play; otherwise II wins.

To show that $L = a^*b(a+b+c)^*$ is not definable, we show that for each r there are words w and v such that $w \in L$ and $v \notin L$, but player II has a winning-strategy in the r-round game on the string models corresponding to w and v. We take

$$\begin{array}{rcl} w & = & a^{r+1}b(a^r ca^r b)^r \\ v & = & a(a^r ca^r b)^r. \end{array}$$

Let $n_i(b, W)$ be the number of b points strictly greater than the distinguished point in W at the end of round i; similarly for $n_i(b, V)$. The winning strategy for player II is to match a b with a b and a c with a c, and to maintain the assertion that either $n_i(b, W) = n_i(b, V)$, or else both numbers are at least $r - i$ (and similarly for c). ❑

Syntax	**Expressions** r	$0 \mid a \mid rs \mid r+s \mid r^+$
Semantics	**Main Clauses**	$[\![r^+]\!] = \bigcup_{n>0} [\![r]\!]^n$, where $[\![r]\!]^1 = [\![r]\!]$ and $[\![r]\!]^{n+1} = [\![r]\!]^n [\![r]\!]$

Figure 9. The syntax and semantics of ε-free regular expressions

Temporal Logic of Strings

We can define all star-free languages if we add the temporal operator $\mathcal{U}$, called *until*. This logic, which we will call PTL, is defined in Figure 8. We can define the language from Proposition 13 in PTL.

EXAMPLE 14. The language $a^*b(a+b+c)^*$ is defined by the following formula

$$\mathcal{U}(b, a)$$

Thus, adding $\mathcal{U}$ gives us a more expressive language. In fact, Etessami and Wilke [26] have shown that there is an "until hierarchy," based on the nesting depth of $\mathcal{U}$. The following theorem characterizes the expressive power of PTL extending the classical characterization of temporal logic by Kamp [44].

THEOREM 15. *The following are equivalent for a language $L \subseteq \Sigma^+$:*

1. *L is FO definable (over the signature $<$ and monadic predicates corresponding to the alphabet letters).*
2. *L is definable in PTL.*
3. *L is star-free.*

Proof. The equivalence (1) iff (3) is due to McNaughton [66]. For an accessible proof, see [25]. The equivalence (1) iff (2) uses Gabbay's [31] separation method, but can also be proved algebraically [19]. See also [30]. ❏

EXAMPLE 16. The language $(aa)^+$ is not PTL definable.

Proof. See [92]. ❏

We will now consider an extension of the star-free languages, called the *regular languages*, defined in Figure 9.

Propositional Dynamic Logic of Strings

The regular languages were first logically characterized by Büchi who showed that they correspond to the languages definable in the monadic second order logic of strings (MSO). We will use propositional dynamic logic (PDL), defined in Figure 10, to characterize them. First, notice that $(aa)^+$ is definable:

EXAMPLE 17. The language $(aa)^+$ is defined by

$$a \wedge \langle\rightarrow\rangle a \wedge [\rightarrow; a?; \rightarrow; a?]^* \neg\langle\rightarrow\rangle\top$$

We will now define the automata theoretic model of the regular languages: finite automata. A *finite automaton* (FA) M is a structure $(\Sigma, Q, F, q_0, \Delta)$ where Σ is a alphabet, Q is a finite set of states, $F \subseteq Q$ is the set of final states, q_0 is the initial state, and Δ is a finite set of transition rules of the form $(q, a) \rightarrow p$ with $a \in \Sigma$ and $p, q \in Q$. We define the transition relation $\Rightarrow_M \subseteq (Q \times \Sigma^*) \times (Q \times \Sigma^*)$ inductively as follows:

$$(q, \varepsilon) \Rightarrow_M (q, \varepsilon)$$
$$(q, aw) \Rightarrow_M (p, w)$$

where $(q, a) \rightarrow p$ is a transition rule in Δ. We say that M *accepts* a string w if $(q_0, w) \Rightarrow_M^* (p, \varepsilon)$ where $p \in F$ and $\Rightarrow_M^*$ is the reflexive, transitive closure of $\Rightarrow_M$. Given an FA M, the language accepted by M, denoted by $L(M)$, is defined as

$$L(G) = \{w \mid (q_0, w) \Rightarrow_M^* (p, \varepsilon), p \in F\}$$

One interesting observation, using the automata theoretic characterization of regular languages, is that they are closed under complementation (notice that "$-$ " is not included in the definition of regular expressions).

Syntax	**Formulas** φ	$p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid [\pi]\varphi$
	Programs π	$\rightarrow \mid ?\varphi \mid \pi;\sigma \mid \pi \cup \sigma \mid \pi^*$
Semantics	**Main Clauses**	$W, i \models [\pi]\varphi$ iff for all j such that $(i, j) \in [\![\pi]\!]_W, W, j \models \varphi$
		$[\![?\varphi]\!]_W = \{(i, i) : i \in [\![\varphi]\!]_W\}$
		$[\![\pi;\sigma]\!]_W = [\![\pi]\!]_W; [\![\sigma]\!]_W$
		$[\![\pi \cup \sigma]\!]_W = [\![\pi]\!]_W \cup [\![\sigma]\!]_W$
		$[\![\pi^*]\!]_W = ([\![\pi]\!]_W)^*$

Figure 10. Propositional Dynamic Logic (PDL) on string models

THEOREM 18. *The following are equivalent for a language $L \subseteq \Sigma^+$:*

1. *L is definable in PDL.*
2. *L is definable in MSO.*
3. *L is regular.*
4. *L is accepted by a FA.*

Proof. The equivalence (2) iff (3) is Büchi's theorem. Again, see [25] for an accessible proof. The equivalence (3) iff (4) is known as Kleene's theorem, see e.g. [39]. For a proof of the equivalence (1) iff (3), see Kracht [53]. ❑

Two interesting results that use algebraic proofs show that it is decidable whether a regular language is definable in $\mathcal{L}(\rightarrow, \rightarrow^*)$ or in PTL [19]. Since it is decidable for any regular language L whether $L = \Sigma^+$, it is decidable whether a formula in MSO is valid over string models [10].

Variations

Other non-modal logics that have been studied in the context of strings include monadic transitive closure (MTC) [3] and least fixed point (MLFP) logic [79], as well as logics with modular counting quantifiers [93]. The latter are of interest because they allow to extend FO logic so that $(aa)^+$ becomes definable without going to the full power of MSO. The logics MTC and MLFP define precisely the regular languages, since it is easy to see that PDL $\leq$ MTC $\leq$ LFP $\leq$ MSO. The equivalence then follows from theorem 18.

Extensions

The first proof that a natural language is not regular was given by Chomsky [16, 17]. Thus, we would have to find stronger logics to describe natural languages within this framework. This is however not the line of research pursued (Rounds [85] being a notable exception), for two reasons. First, decidability of the logical formalism employed is of some importance, as this line of research ultimately aims to contribute to computational linguistics. However, trying to find *decidable* extensions of PDL or MSO is quite challenging (a point we will revisit later). There exists a characterization of the CFLs in terms of an extension of MSO [58], allowing quantification over special kinds of binary relations, so-called "matchings". Even though the question whether this logic is decidable is not addressed there, validity is undecidable for any logic characterizing CFLs, since the questions whether $L(G) = \Sigma^+$ is undecidable for CFGs. Such logics are also bound to be odd, since CFLs are not closed under complementation. The second problem is that using a logics for *strings* does not give a notion of strong generative capacity: the process of verifying that a formula is true of a string does not assign a structure to that string, as the process of deriving a string using a grammar does.

Digression

One point which should be of interests to modal logicians concerns the sense in which *words* are like *(modal) sentences*. The analogy is neatly captured in coalgebra (see Chapter 6), especially in studies pertaining to *coalgebraic logic*. We are considering several functors on the category of sets. Here $\mathcal{P}$ is the power set functor, and $\mathcal{P}_f$ is the finite power set functor. *AtProp* is a set of atomic propositions, and A is an alphabet. We have the following analogies,

	Kripke semantics	automata
Functor $F(x)$ on sets	$\mathcal{P}(x) \times \mathcal{P}(AtProp)$	deterministic : $x^A \times \{0,1\}$ non-deterministic : $\mathcal{P}_f(x)^A \times \{0,1\}$
coalgebra	Kripke model	deterministic automaton
final coalgebra	canonical model	regular languages
notion of equivalence	bisimulation	bisimulation.

Kripke models, deterministic, and non-deterministic automata are described as coalgebras of the given functors. In both cases, the elements of the carrier of the coalgebra may be thought of as *states*. (However, coalgebras do not include specified "real worlds" or "start states".) In the case of the automata, the state sets might be infinite, but our use of $\mathcal{P}_f$ insures that they will be finitely branching. The set $\{0,1\}$ in the automata functors is there to equip the state set with *accepting* and *non-accepting* states. The final

coalgebra in each case turns out to be an important mathematical object, and the reader can see the sense in which the canonical model is the analog of the regular languages. Indeed, modal sentences might be thought of as the record of "possible observations" on Kripke models in the same way that words are on automata. Finally, from coalgebra we have a very general notion of equivalence, the coalgebraic bisimulation. The special cases of this are bisimulation of Kripke models and also the bisimulation of automata (the largest such is the Myhill-Nerode equivalence relation). See Rutten [86] for more information on the connection between automata and coalgebra.

3.3 Logics of trees

Since extending the logic of strings to capture more complex string languages than the regular languages often leads to undecidability, one approach to extending the coverage of our logic is to describe more complex structures: move from strings to trees. Thus, the Kripke structures we will be considering are trees, and the logics will contain more complicated modalities to describe trees. One immediate advantage of this approach for linguistic purposes is that these logics will automatically be connected to strong generative capacity, since they describe sets of trees. One disadvantage is that the *recognition* or *parsing* problem, which in the string case just amounts to model checking, now involves satisfiability checking (see below).

The extension of the descriptive approach to trees was originally also motivated by decidability questions [97]. Even though the connections to CFLs were pointed out by Thatcher [95], this line of research did not find applications in linguistics until the development of constraint based grammar formalisms which replaced the derivational approach to natural language syntax. The work of Rogers [83], Kracht [54], and others provided formal models for these constraint based grammar formalisms and established formal language theoretic results for them at the same time.

As mentioned above our Kripke structures will now be trees. We will use the concept of tree domains [35] to define such Kripke structures. A (finite, binary) *tree domain*, T, is a finite subset of $\{0,1\}^*$, such that for all $u, v \in \{0,1\}^*$

1. if $uv \in T$, then $u \in T$, and

2. if $u1 \in T$, then $u0 \in T$.

A string in T describes a path from the root to a node, where 0 means "go left" and 1 means "go right". We identify nodes with the path leading to them. Thus, ε is the root. The first condition above says that if there is a path to a node, then there is a path to any node above it (this is called prefix closure). The second condition says that if a node has a right daughter, then it has a left daughter (called left sibling closure).

The main relations between nodes in a tree that are of interest in linguistics are domination and linear precedence. We say that that a node $u \in T$ *dominates* a node $v \in T$ if for some $w \in \{0,1\}^*$, $v = uw$. A special case of domination is the parent-of relation, defined by: u is the parent of v if $v = u0$ or $v = u1$. We say that u *linearly precedes* v if for some $x, y, z \in \{0,1\}^*$, $u = x0y$ and $v = x1z$. Following Rogers [83], we will denote the domination relation by $\triangleleft^*$, the parent-of relation by $\triangleleft$, and linear precedence by $\prec$. Thus, our Kripke frames will be variations of the form $(T, \triangleleft, \triangleleft^*, \prec)$, where T is a tree domain.

Regular tree languages

In order to generalize from strings to labeled trees, we will now consider *ranked alphabets* in which each symbol has an arity or rank. For surveys of tree languages see Gécseg and Steinby [34] or Thatcher [96]. Let Σ be a ranked alphabet. We will denote the set of n-ary symbols in Σ by Σ_n. The set of terms over Σ is denoted by T_Σ. A subset of T_Σ is called a *tree language*.

In a number of settings, trees are considered to be labeled with boolean features, rather than with ranked symbols. We note that these two approaches commensurable using the following representation. Given a finite set of boolean features $F = \{f_1, \ldots, f_n\}$, the *binary ranked alphabet based on* F, Σ^F, is defined as

$$\Sigma^F = \{f_1, \neg f_1\} \times \cdots \times \{f_n, \neg f_n\} \times \{0, 2\}$$

where each $f_i, \neg f_i$ represents whether or not a feature holds at a given node and 0 or 1 represent the arity of the symbol. Thus, $(f_1, \neg f_2, 0)$ would be a leaf symbol, and $(f_1, \neg f_2, 2)$ would be an internal node symbol. The previous definition can be easily generalized to trees of any arity.

The *yield* of a tree, t, is the string over Σ_0 which is obtained by concatenating the symbols at the leaves of t from left to right, or more formally:

$$\begin{aligned} \mathit{yield}(c) &= c, \text{ for } c \in \Sigma_0 \\ \mathit{yield}(f(t_1, \ldots, t_n)) &= \mathit{yield}(t_1) \ldots \mathit{yield}(t_n), \text{ for } f \in \Sigma_n \end{aligned}$$

A (bottom-up, non-deterministic) *finite tree automaton* (FTA) M is a structure of the form (Σ, Q, F, Δ) where Σ is a ranked alphabet, Q is a finite set of states, $F \subseteq Q$ is the set of final states, and Δ is a finite set of transition rules of the form $f(q_1, \ldots, q_n) \to q$ with $f \in \Sigma_n$ and $q, q_1, \ldots, q_n \in Q$. An FTA is *deterministic* if there are no two transition rules with the same left-hand-side. It can be shown that the bottom-up variety of finite tree automata can be determinized, while the top-down variety cannot.

A *context* s is a term over $\Sigma \cup \{x\}$ containing the zero-ary term x exactly once. We write $s[x \mapsto t]$ for the term that results from substituting x in s with t. Given a finite tree automaton $M = (\Sigma, Q, F, \Delta)$ the derivation relation $\Rightarrow_M \subseteq T_{Q\cup\Sigma} \times T_{Q\cup\Sigma}$ is defined by $t \Rightarrow_M t'$ if for some context $s \in T_{\Sigma\cup Q\cup\{x\}}$ there is a rule $f(q_1, \ldots, q_n) \to q$ in Δ, and

$$\begin{aligned} t &= s[x \mapsto f(q_1, \ldots, q_n)] \\ t' &= s[x \mapsto q] \end{aligned}$$

We use $\Rightarrow_M^*$ to denote the reflexive, transitive closure of $\Rightarrow_M$. A finite automaton M *accepts* a term $t \in T_\Sigma$ if $t \Rightarrow_M^* q$ for some $q \in F$. The *tree language accepted* by a finite tree automaton M, $L(M)$, is

$$L(M) = \{t \in T_\Sigma \mid t \Rightarrow_M^* q, \text{ for some } q \in F\}.$$

A tree language, L, is *regular* if $L = L(M)$ for some FTA M.

The following example is concerned with the Circuit Value Problem (CVP), in which the trees labeled with boolean functions are evaluated. It is interesting to note that a number of separation results of logically defined tree languages use trees labeled with boolean functions [79].

EXAMPLE 19. Let $\Sigma = \{\wedge, \vee, 0, 1\}$. The tree language $CVP \subseteq T_\Sigma$ such that each tree in CVP evaluates to true can be accepted by the following FTA, $M = (\Sigma, Q, F, \Delta)$, where

$$\begin{aligned} Q &= \{t, f\} \\ F &= \{t\} \end{aligned}$$

and

$$\Delta = \left\{ \begin{array}{ll} 0 \rightarrow f, & 1 \rightarrow t, \\ \wedge(t,t) \rightarrow t, & \wedge(t,f) \rightarrow f, \\ \wedge(f,t) \rightarrow f, & \wedge(f,f) \rightarrow f, \\ \vee(t,t) \rightarrow t, & \vee(t,f) \rightarrow t, \\ \vee(f,t) \rightarrow t, & \vee(f,f) \rightarrow f \end{array} \right\}$$

Given a finite sets of feature $F = \{f_1, \ldots, f_n\}$ and a feature $f_i \in F$, we define the *projection*, π, that eliminates f_i in the natural way:

$$\pi : \Sigma^F \rightarrow \Sigma^{F-\{f_i\}}$$

This definition can be extended to arbitrary subsets $G \subseteq F$, where

$$\pi : \Sigma^F \rightarrow \Sigma^{F-G}$$

Given a projection $\pi : \Sigma^F \rightarrow \Sigma^{F-G}$, we extend π to a tree homomorphism $\hat{\pi} : T_{\Sigma^F} \rightarrow T_{\Sigma^{F-G}}$ as follows:

$$\begin{aligned} \hat{\pi}(c) &= \pi(c) \\ \hat{\pi}(f(t_1, \ldots, t_n)) &= \pi(f)(\hat{\pi}(t_1), \ldots, \hat{\pi}(t_n)) \end{aligned}$$

with $c \in \Sigma_0$ and $f \in \Sigma_n, n > 0$. For a tree language L, we define $\hat{\pi}(L) = \{\hat{\pi}(t) \mid t \in L\}$.

We will consider the relationship between regular tree languages and the derivation trees of CFGs.

PROPOSITION 20. *(Thatcher [95]) If $L \subseteq T_\Sigma$ is a regular tree language, then*

$$\{yield(t) \mid t \in L\}$$

is a CFL.

While the yields of regular tree languages are CFLs, regular tree languages are more complex than the derivation trees of CFG. In order to compare the regular tree languages to the derivation trees of CFGs, we formalize the latter using the local tree languages.

The *fork* of a tree t, *fork*(t), is defined by

$$\begin{aligned} fork(c) &= \varnothing \\ fork(f(t_1, \cdots, t_n)) &= \{(f, root(t_1), \ldots, root(t_n))\} \cup \bigcup_{i=1}^{n} fork(t_i) \end{aligned}$$

with $c \in \Sigma_0$, $f \in \Sigma_n, n > 0$, and *root* being the function that returns the symbol at the root of its argument. For a tree language L, we define

$$\mathit{fork}(L) = \bigcup_{t \in L} \mathit{fork}(t)$$

The intuition behind the definition of *fork* is that an element of $\mathit{fork}(T_\Sigma)$ corresponds to a rewrite rule of a CFG. Note that $\mathit{fork}(T_\Sigma)$ is always finite, since Σ is finite.

A tree language $L \subseteq T_\Sigma$ is *local* if there are sets $R \subseteq \Sigma$ and $E \subseteq \mathit{fork}(T_\Sigma)$, such that, for all $t \in T_\Sigma, t \in L$ iff $\mathit{root}(t) \in R$ and $\mathit{fork}(t) \subseteq E$.

We quote without proof the following two theorems by Thatcher [95].

THEOREM 21. *(Thatcher [95]) A tree language is a set of derivation trees of some CFG iff it is local.*

THEOREM 22. *(Thatcher [95]) Every local tree language is regular.*

While there are regular tree languages that are not local, the following theorem, also due to [95], demonstrates that we can obtain the regular tree languages from the local tree languages via projections. We will review the main points of the proof, because we will use some of its details later on.

THEOREM 23. *(Thatcher [95]) For every regular tree language L, there is a local tree language L' and a one-to-one projection π, such that $L = \hat{\pi}(L')$.*

Proof. Let L be a regular tree language. Assume that L is accepted by the deterministic FTA $M = (\Sigma, Q, F, \Delta)$. We define L' terms of R and E as follows: $R = \Sigma \times F$ and

$$E = \{((f, q), (f_1, q_1), \ldots, (f_n, q_n)) \mid f(q_1, \ldots, q_n) \to q \in \Delta, f_1, \ldots, f_n \in \Sigma\}$$

We then define $L' = \{t \in T_{\Sigma \times Q} \mid \mathit{root}(t) \in R, \mathit{fork}(t) \subseteq E\}$. Notice that the trees in L' encode runs of M. The tree homomorphisms $\hat{\pi}$ based on the projection $\pi : \Sigma \times Q \to \Sigma$ maps L' to L as can be easily verified.

It should be noted that, since M is deterministic, there exists exactly one accepting run for each tree in $L(M)$ and thus the homomorphism $\hat{\pi} : L' \to L$ is one-to-one. ❑

This rather technical result is of some importance in the context of linguistic application, for it implies that we can use frameworks of lower complexity to describe the same structures as a more complex framework *if we use more complex categories or features.* Since we can also add new categories as names for the more complex ones, we can use a less complex framework to describe the same structures as a more complex framework by adding more categories. Thus, parsimony would seem to imply that we should always use the simpler framework. However from the point of linguistics, the use of complex or additional features needs to be justified. To further elaborate on the previous point, we will have to keep in mind that all of the logics we will consider can define the local tree languages and all the languages they can define are regular. Thus undefinability will always mean undefinability over a fixed finite set of propositional variables, since we can always define a regular, undefinable tree language by using more features.

The basic modal logic of trees: $\mathcal{L}_{core}$

To the best of our knowledge, the first explicit use of modal logic to define tree languages can be found in [7]. Two variations of this logic were considered in [8, 9], of which we

Syntax	**Formulas** φ	$p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid [\pi]\varphi$
	Programs π	$\rightarrow \mid \leftarrow \mid \uparrow \mid \downarrow \mid \pi^*$
Semantics	**Main Clauses**	$[\![\rightarrow]\!]_T = \{(u0, u1) \mid u1 \in T\}$
		$[\![\leftarrow]\!]_T = \{(u1, u0) \mid u1 \in T\}$
		$[\![\downarrow]\!]_T = \{(u, ui) \mid i \in \{0,1\}, ui \in T\}$
		$[\![\uparrow]\!]_T = \{(ui, u) \mid i \in \{0,1\}, ui \in T\}$

Figure 11. Modal logic of trees: $\mathcal{L}_{core}$

will consider the latter. The basic modal logic of trees, $\mathcal{L}_{core}$, is defined in Figure 11. Again, we say that a tree T satisfies a formula φ if $T, \varepsilon \models \varphi$. A language L is *definable* in this (or another language) if there is a sentence φ so that L is exactly the set of trees satisfying a sentence φ.

The following proposition establishes that $\mathcal{L}_{core}$ is expressive enough to define any binary branching, local tree language. The restriction to binary branching is only due to the fact that we defined our tree domains to be binary branching.

PROPOSITION 24. *Let $L \subseteq T_\Sigma$ be a local tree language. There is a sentence φ_G in $\mathcal{L}_{core}$ that defines L.*

Proof. By Theorem 21, there is a CFG G such that L is equal to the derivation trees of G. Let $G = (\Sigma, \Gamma, P, S)$. Since we are only considering binary branching trees, every rule in P is of the form $A \rightarrow BC$ or $A \rightarrow a$ with $A, B, C \in \Gamma$ and $a \in \Sigma$. We can simply encode the rules directly in our logic:

$$A \rightarrow \bigvee_{A \rightarrow BC \in P} \langle\downarrow\rangle(B \wedge \langle\rightarrow\rangle C)$$

and

$$A \rightarrow \bigvee_{A \rightarrow a \in P} (\langle\downarrow\rangle a)$$

This ensures that the models of φ_G are parse trees of G. However, we further need to ensure only the parse trees of G model φ_G. So, we need to express that each node makes exactly one symbol true:

$$[\downarrow^*]\Big(\bigvee_{a \in (\Sigma \cup \Gamma)} a \wedge \bigwedge_{a \neq b} (\neg a \vee \neg b)\Big)$$

that the start symbol of the grammar is true at the root: S, that the terminal symbols are true at the leaves:

$$[\downarrow^*]\Big(\bigvee_{a \in \Sigma} a \rightarrow \neg\langle\downarrow\rangle\top\Big)$$

and that the non-terminal symbols are true at the internal nodes

$$[\downarrow^*]\Big(\bigvee_{A \in \Gamma} A \rightarrow \langle\downarrow\rangle\top\Big)$$

❑

As is observed by Blackburn and Meyer-Viol, this translation of a CFG into logical formulas brings with it a change in perspective. Instead of a *procedural* or *derivational* perspective that considers CFG rules to be rewrite rules, we move to a *declarative* or *descriptive* perspective that considers CFG rules to be *constraints*. This change in perspective is the main motivation for the application of logic in syntax, because of a similar change in perspective that occurred in a number of grammar formalisms proposed by linguists in the 1980s, most notably Chomsky's "Government and Binding" (GB) [18] and Gazdar, Klein, Pullum, and Sag's "Generalized Phrase Structure Grammar" (GPSG) [33].

ID/LP Grammars

The rules of a CFG encode two kinds of information: the categories of a node and its children, and the order in which the categories of the children occur. Thus, a rule of the form $A \to BC$ tells us that a node labeled A can have two children, one labeled B, the other C, and that the node labeled B precedes the node labeled C. Linguists have observed that separating these two notions can lead to more compact grammars. Thus, ID/LP grammars have been proposed that consist of *unordered* rewrite (immediate dominance or ID) rules, $A \to B, C$, and *linear precedence* (LP) rules, $B < C$. Linear precedence rules only apply to sisters, which is why we used $<$ rather than $\prec$ which applies to arbitrary nodes.

ID/LP grammars can be very naturally expressed in $\mathcal{L}_{core}$; in fact ID/LP grammars are, in some sense, a very limited logic for trees. See Gazdar et al. [33] or Shieber [88] for applications and detailed examinations of ID/LP grammars.

Variations of $\mathcal{L}_{core}$

Two additional basic modal logics of trees have been considered by Blackburn and associates [7, 8]. The first includes the connectives $\varphi \Rightarrow \psi$ and $\bullet(\varphi_1, \ldots \varphi_n)$. The latter is used in the context of trees with n children, so we will only consider the case where n is 2. Their semantics are given by $T, v \models \varphi \Rightarrow \psi$ iff for all u, $T, u \models \varphi \to \psi$, and $T, v \models \bullet(\varphi, \psi)$ iff $T, u0 \models \varphi$ and $T, u1 \models \psi$. Notice that the purpose of $\bullet$ is to combine immediate dominance and linear precedence into one connective.

Blackburn and Meyer-Viol [8] define a modal logic of trees that differs from $\mathcal{L}_{core}$ in that it contains modalities for the left and right daughter: $\downarrow_1, \downarrow_2$.

Temporal Logic of Trees

We now move on to an extension of $\mathcal{L}_{core}$, temporal logic. The syntax and semantics of propositional tense logic on trees, $\mathcal{X}_{until}$, is defined in Figure 12. The main application of $\mathcal{X}_{until}$ was given by Palm [70], though with a different formulation which we will consider below. We follow here the formulation of Marx [63], because it lends itself to a more direct proof of equivalence with FO.

THEOREM 25. *[63] The following are equivalent for a tree language $L \subseteq T_\Sigma$:*

1. *L is FO definable.*

2. *L is definable in $\mathcal{X}_{until}$.*

Syntax	$p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathcal{U}_{\rightarrow}(\varphi,\psi) \mid \mathcal{U}_{\leftarrow}(\varphi,\psi) \mid \mathcal{U}_{\uparrow}(\varphi,\psi) \mid \mathcal{U}_{\downarrow}(\varphi,\psi)$
Semantics	$T, u \models \mathcal{U}_{\downarrow}(\varphi,\psi)$ iff there exists a v such that $u \lhd^* v$, $T, v \models \varphi$, and for all w such that $u \lhd^* w \lhd^* v$, $T, w \models \psi$

Figure 12. Temporal logic of trees: $\mathcal{X}_{until}$ (only one clause in the semantics)

Syntax	**Formulas** φ	$p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid [\pi]\varphi$
	Programs π	$\rightarrow \mid \leftarrow \mid \uparrow \mid \downarrow \mid \pi_\varphi \mid \pi^*$
Semantics	**Main Clauses**	$[\![\pi_\varphi]\!]_T = \{(u,v) \mid (u,v) \in [\![\pi]\!]_T, T, u \models \varphi\}$

Figure 13. Conditional path logic of trees: $\mathcal{L}_{cp}$

While the notion of regular expressions can be generalized to trees, the correspondence between star-free expressions and FO (or $\mathcal{X}_{until}$) definability breaks down at this level. In fact, Thomas and Potthoff [81] showed that every regular language that does not contain unary branching symbols is star-free. The question whether FO definability of regular tree language is decidable is still open.

Variations of $\mathcal{X}_{until}$

As was mentioned above, Palm's [70] application of $\mathcal{X}_{until}$ was carried out using a different formulation which he called propositional tense logic and which Afanasiev et al. [1] called conditional path logic, $\mathcal{L}_{cp}$. The syntax and semantics of $\mathcal{L}_{cp}$ are defined in Figure 13.

X-bar theory

As was mentioned above, which non-terminals are used in a natural language grammar matters to linguists. The point again is that the label assigned to a node in a tree signifies the grammatical category of the constituent it dominates. One theory of the organization of non-terminals and their rules is X-bar theory, which provides the foundation for a variety of grammar formalisms, including GB and GPSG. There are many variations of X-bar theory, so the particular formulation discussed here may not agree with those found in other places.

In terms of the organization of the non-terminals of a grammar, X-bar theory stipulates that there is a finite set of *lexical categories*, like $N(oun)$, $V(erb)$, $P(reposition)$, $A(djective)$, $Adv(erb)$, corresponding to the parts of speech, and that all other non-terminals are *projections* of the lexical categories. The idea of a projection is best motivated by the following example. The constituent *tall man* consists of two words, a noun and an adjective. When considering what the category of the constituent should be, we should take into account that *tall man* behaves more like a noun than like an adjective, which can be verified by substituting *tall man* for a noun in a sentence, preserving grammaticality, and substituting it for an adjective in a sentence, not preserving grammaticality. Thus, the category of *tall man* should be derived from the category of *man*. The category that X-bar theory assigns to the phrase is called N' (pronounced N-bar). N' is a projection of N. While X-bar theory within GB considered N and N' as atomic categories, the idea that the bar-level of a node is a syntactic feature is due to GPSG.

While there are various proposal for X-bar theory, we will assume that all rules of an X-bar grammar should be of the form

$$X'' \to X', Y'' \tag{11}$$

$$X' \to X', Y'' \tag{12}$$

$$X' \to X, Y'' \tag{13}$$

The non-terminal Y'' has different roles in the three rule schemata, each of which has a name in X-bar theory. In rule schema (11), Y'' is called the *specifier*; in rule schema (12), it is called the *adjunct*, and in rule schema (13), it is called the *complement*. In each of the rules, the X or X' on the right hand side is called the *head*.

It has been observed in a variety of contexts [48, 51, 70] that it is desirable to dispense with the bar-feature and to define the constraints posed by the X-bar schemata in terms of projections. Thus, we would like to define a constraint that states that every node has a path to a leaf such that the node, the leaf, and all the nodes on the path have the same lexical features. This can expressed in $\mathcal{L}_{cp}$ as follows. First, we state that a feature φ belongs to a head:

$$hd\ \varphi \equiv \varphi \wedge head$$

Then, we state that a feature φ is projected from a leaf:

$$proj\ \varphi \equiv \langle \downarrow^*_{hd\ \varphi} \rangle (hd\ \varphi \wedge leaf)$$

Finally, we can restate the X-bar convention by requiring every node to be a projection, given a finite set of lexical features Lex:

$$[\downarrow^*](\bigvee_{\varphi \in Lex} proj\ \varphi)$$

Notice that we would need a feature to indicate that a node is the head in case two siblings share the same lexical feature. Furthermore, there are certain regularities that this head feature has to observe, such as that no two sisters may both be heads:

$$[\downarrow^*](head \to \neg(\langle \leftarrow \rangle head \vee \langle \rightarrow \rangle head))$$

Dynamic Logic of Trees

The first descriptive characterization of the regular tree languages was obtained by Doner [23], and Thatcher and Wright [97]. They generalized Büchi's theorem to trees.

THEOREM 26. *The following are equivalent for a tree language $L \subseteq T_\Sigma$:*

1. *L is regular.*
2. *L is definable in MSO.*

Kracht [49] introduced PDL on trees in the context of model theoretic syntax.

While the correspondence between $\mathcal{X}_{until}$ and FO continues to hold in the generalization from strings to trees, the same is not true for the correspondence between PDL and MSO on strings, as was shown by Kracht, a topic we shall investigate in detail in the next section.

Syntax	**Formulas** φ	$p_i \mid \neg\varphi \mid \varphi \wedge \psi \mid [\pi]\varphi$
	Programs π	$\rightarrow \mid \leftarrow \mid \uparrow \mid \downarrow \mid ?\varphi \mid \pi;\sigma \mid \pi \cup \sigma \mid \pi^*$

Figure 14. Dynamic logic of trees

Undefinability: Inessential Features

The relationships between the three logics discussed above are well-understood, in that $\mathcal{L}_{core}$ is properly included in $\mathcal{X}_{until}$, which is properly included in PDL, which in turn is properly included in MSO. There is a central property that can be used to describe the languages that can be defined in one logic, but not in another. This property was first introduced by Kracht [50] and it is defined in terms of *inessential features*.

Let F be a finite set of features, $G \subseteq F$, $L \subseteq T_{\Sigma^F}$, and $\pi : \Sigma^F \rightarrow \Sigma^{F-G}$ be a projection. We call the features in G *inessential for* L if the homomorphism $\hat{\pi} : L \rightarrow T_{\Sigma^{F-G}}$ based on π is one-to-one. The intuition for this definition of inessential features is that no two trees in L can be distinguished using features in G. Thus, given a tree t in $\hat{\pi}(L)$, we can recover the features from G in t using $\hat{\pi}^{-1}$, since $\hat{\pi}$ is one-to-one.

EXAMPLE 27. The bar feature of the version of X-bar theory sketched above is inessential. To see that, notice that there is only one head (bar-level 0) which has a maximal projection (bar-level 2) and all projections in between are of bar-level 1.

While being an inessential feature is defined with respect to a language, being eliminable is defined with respect to a logic and a language. Let F be a finite set of features, $G \subseteq F$, $L \subseteq T_{\Sigma^F}$, $\pi : \Sigma^F \rightarrow \Sigma^{F-G}$ be a projection, and $\mathcal{L}$ be a logic. Suppose that L is definable in $\mathcal{L}^F$. We say that G is *eliminable in* $\mathcal{L}$ *for* L if $\hat{\pi}(L)$ is definable in $\mathcal{L}^{F-G}$.

It should be noted that this definition of eliminability does not coincide with Kracht's [50], who defines eliminable as being globally explicitly definable. Kracht's definition implies the definition used here, and thus is stronger. However, since we are interested in *ineliminability*, by contraposition, the definition employed here implies Kracht's definition of ineliminability.

The following, well-known, inclusions follow primarily from the definition of the three modal logics.

THEOREM 28. $\mathcal{L}_{core} \leq \mathcal{L}_{cp} \leq PDL_{tree} \leq MSO$

Proof. The first two inclusions follow from the definitions of these logics. The third inclusion follows from the fact that transitive closure is MSO-definable. ❑

Next, we consider strictness of these inclusions.

PROPOSITION 29. *[87] Let* $F = \{a, b\}$*. The tree language* $L_1 \subseteq T_{\Sigma^F}$ *such that each tree in* L_1 *contains a path from the root to a leaf at which exactly one* a *holds is not* $\mathcal{L}_{core}$*-definable, but is* $\mathcal{L}_{cp}$*-definable.*

PROPOSITION 30. *Let* $\Sigma = \{\wedge, \vee, 0, 1\}$*. The tree language* $CVP \subseteq T_\Sigma$ *such that each tree in* CVP *evaluates to true is not* $\mathcal{L}_{cp}$*-definable, but is* PDL_{tree}*-definable.*

Proof. Potthoff [80] showed that CVP is not definable in an extension of first-order logic with modular counting quantifiers, and since $\mathcal{L}_{cp}$ is equivalent to first-order logic on trees [1], the undefinability follows. That CVP is definable in PDL$_{tree}$ is shown in [1]. ❑

PROPOSITION 31. *[52, 53] Let $F = \{p, q\}$. Let $L_2 \subseteq T_{\Sigma^F}$ where each tree in L is a ternary branching tree such that p is true along a binary branching subtree and q is true at all leaves at which p is true. The language $L_3 \subseteq T_{\Sigma^{\{q\}}}$ obtained from the projection that eliminates p is not PDL_{tree}-definable, but is MSO-definable.*

These three propositions demonstrate the strictness of the inclusion of the three modal logics and MSO. Next, we will consider how languages that are undefinable in one of these logics can be defined with additional features.

THEOREM 32. *[102] There exists a set of features F, a tree language $L \subseteq T_{\Sigma^F}$, and a subset $G \subseteq F$, such that G is ineliminable in $\mathcal{L}_{core}$ (resp. $\mathcal{L}_{cp}$) but eliminable in $\mathcal{L}_{cp}$ (resp. PDL_{tree}).*

Proof. Both of these constructions work the same way. Given two of our logics $\mathcal{L}_1, \mathcal{L}_2$, with $\mathcal{L}_1 < \mathcal{L}_2$, pick a tree language, L, that is not definable in $\mathcal{L}_1$ but is definable in $\mathcal{L}_2$, which exists by propositions 29 and 30.

By Theorem 28, we know that L is regular, and by Theorem 24, we know that any local tree language is definable in $\mathcal{L}_1$. Given a deterministic FTA $M = (\Sigma, Q, F, \Delta)$, with $L = L(M)$, we can use theorem 23 to construct a local tree language $L' \subseteq T_{\Sigma \times Q}$ such that $\hat{\pi}(L') = L$. Now, the features in Q are inessential, since M is deterministic, but ineliminable, since L is undefinable in $\mathcal{L}_1$. However, since L is definable in $\mathcal{L}_2$, the features in Q are eliminable in $\mathcal{L}_2$. ❑

The previous theorem can be strengthened in that it can be used to *characterize* the tree languages that are undefinable in some logic $\mathcal{L}_1$ but definable in some other logic $\mathcal{L}_2$, with $\mathcal{L}_1 \leq \mathcal{L}_2$.

THEOREM 33. *[102] Any tree language that is not definable in $\mathcal{L}_{core}$ (resp. $\mathcal{L}_{cp}$) but is definable in $\mathcal{L}_{cp}$ (resp. PDL_{tree}) can be defined with additional, inessential features in $\mathcal{L}_{core}$ (resp. $\mathcal{L}_{cp}$) that are not eliminable in $\mathcal{L}_{core}$ (resp. $\mathcal{L}_{cp}$).*

Model Theoretic Syntax and Parsing

Recall that we generalized from strings to trees because we wanted to retain decidability and because we wanted to have a formalism that associates grammatical structure to an unstructured string. While decidability has been retained by this move, we need to say a little bit about how model theoretic syntax associates structures with strings. It should be noted that CFGs are formalisms that generate strings and that the structures that they assign to the strings arise in the process of generating the string, i.e. that trees are not a primary but a derived notion for formal grammars, Tree Adjoining Grammars being a notable exception. It should also be noted that, in our move to logics of trees, strings are no longer a primary notion because we are talking about trees directly. However, when we are interested in, say, checking whether a particular sentence is grammatical, we are given a string. So, while parsing, the process of determining whether a given grammar generates a given string, for CFG amounts to checking whether the grammar generates the string, this is not quite as straightforward here. The following quote from [2] gives an outline of how parsing in the logical framework might look like:

> The intent here is to translate a given grammar G into a formula φ_G such that the set of trees generated by the grammar is exactly the set of trees that

satisfy φ_G. Parsing, then, is just identifying the set of models of φ_G that yield a given string.

Following an idea proposed by Cornell [20] in the context of parsing with finite tree automata, we can improve on the above parsing procedure by observing that we can describe the set of all trees that yield a given string w, φ_w, and then simply check whether $\varphi_w \wedge \varphi_G$ is satisfiable. Notice, though, that having moved from logics of strings to logic of trees entails that the complexity of parsing, which in the string case is that of *model checking*, now is that of *satisfiability checking*. For all of the modal logics considered here, satisfiability checking is EXPTIME-complete. This is still significantly better than MSO or even FO with $\triangleleft^*$ both of which are non-elementary. However, model checking for the modal logics considered here is linear. For another approach to parsing and model theoretic syntax, see Palm [71].

Variations

Just as in the case of strings, monadic transitive closure (MTC) and least fixed point (MLFP) logic and logics with modular counting quantifiers have been considered on trees [79], as well as Thomas' chain and anti-chain logics [98]. While, over trees, MLFP is equally expressive as MSO, the question whether this equivalence also holds for MTC is currently open.

Kracht [54] also considers a modal logic with quantifiers ranging over propositions which is equivalent to MSO over trees.

Extensions

While the fact that natural languages are not regular has been known since the 1950s, examples of non-context-free phenomena in natural languages were only found in the 1980s; see Shieber [89]. Thus, we again need to consider how to strengthen the logics employed here if we want this approach to be applicable to all natural languages.

One approach, a generalization of the logical characterization of CFLs to trees, is Langholm's [56] characterization of the indexed languages by an extension of MSO which generalizes the logical characterization of CFLs mentioned above to trees. The indexed languages are located strictly between the CFLs and the context-sensitive languages. However, as was pointed out above, since parsing with tree logics involves testing for satisfiability rather than model checking, using an undecidable logic makes this approach uninteresting to computational linguistics.

Other approaches to extending model theoretic syntax to non-regular tree languages include Rogers' [84] extension of MSO to n-dimensional trees and the approach by Mönnich and colleagues [47] that encodes non-regular tree language in regular tree languages. Both approaches have in common that they introduce a new level of abstraction, since the direct connection between a logical formula and the tree it encodes is only available via a translation, which is explicit only in the latter approach. While this move from trees to more complex structures is analogous to the move from strings to trees, the latter move still corresponds to structures employed by linguists (derivation trees) while the former does not. However, both approaches retain decidability. Whether decidable, non-regular extensions of PDL can be used to define interesting classes of tree languages is, at present, an open problem.

3.4 Assessment: why modal logic for syntax and which one?

The foregoing multitude of tree logics raises two questions: what are the advantages and disadvantages of modal logics over classical logics for the description of trees, and similarly between the different modal logics? With respect to classical logic, the advantage is not, as in the general case, that modal logics are decidable while classical logic is not, since even MSO over trees is decidable. However, there is an advantage in complexity: all the modal logics considered are EXPTIME-complete [1], while MSO and FO with $\triangleleft^*$ are not elementary. One exception is FO with two successors, S_1, S_2 which is elementary [27], but not very expressive, since not even $\triangleleft^*$ is FO definable from S_1, S_2. For further discussions of complexity theoretic aspects of MSO, see [61].

Another more general question: why should logic be used at all to formalize grammatical theories? The first advantage that the approach outlined in this chapter has is that it connects a descriptive approach to grammars to a procedural approach: grammars formalized in these logics can be translated into tree automata which can be implemented. Another issue has to do with methodology in linguistics. While some linguists have become downright hostile towards formalization, the methodological paradigm of Government and Binding theory was to formulate more and more "principles;" i.e., general statements about the structure of sentences that were supposed to be true for all languages. However, it was quite unclear how one would check whether or not any new principle was consistent with all the previously stated principles. Formalizing principles from GB in one of these logics would allow to check whether an adding a given principle would make a particular theory contradictory. For further discussions of methodological issues in GB, see Hintikka and Sandu [38].

4 CONCLUSION AND OPEN PROBLEMS

Like other areas of applied mathematics which use formal tools to model phenomena under consideration, logic in general, and modal logic in particular, is one of the main tools for modeling in mathematical linguistics. As we have seen, modal logic is used in semantics to give a formal model of the meanings of the object language, while it is used in syntax to formalize grammatical theories; i.e., the meta-language. While the use of logic in semantics has considerable history with many significant successes, the logical approach to syntax outlined here is relatively new, although its foundations date back further.

There are many applications of logic in linguistics that we have not discussed here, however, two stand out because they contain applications of modal logic: categorial grammar and feature structures. However, both of these topics have already received authoritative surveys in the *Handbook of Logic & Language* [4].

One area of research in mathematical linguistics that has had considerable success in recent years has been the study of learnability of grammar formalisms, particularly of variations of categorial grammars; see Buszkowski [12]. Similar results for model theoretic syntax have not been obtained yet. While there exist interesting approaches to learning logical theories [62] which would seem to be relevant to extending learnability theory to model theoretic syntax, these approaches depend heavily on properties of their main tool, first-order logic. Thus, a significant amount of groundwork would have to be done before one could extend this approach to model theoretic syntax.

Further open problems in model theoretic syntax include computational implementations, for which some progress has already been made by the existing implementations of monadic second order logic [46]. However similar implementations of modal logics of trees or applications of the existing applications to linguistic problems do not seem to exist. The relationship between the different approaches to extending model theoretic syntax to non-regular tree languages outlined above is also currently open. For example, is there an easy way to translate between Rogers' extension in [84] of MSO to n-dimensional trees and the approach by Mönnich and colleagues [47] that encodes non-regular tree language in regular tree languages? Finally, while the different modal logics in this chapter were separated using the tree languages in Propositions 29, 30 and 31, it would be interesting to find linguistically motivated tree languages that can also separate these logics. Until such examples are found, very little motivation seems to exist to use the more expressive logics.

One interesting property that the logical approaches to both syntax and semantics outlined here have in common is that extending their empirical scope to different natural language phenomena depends on corresponding coverage of these phenomena in some syntactic theory. Since it is the main aim of model theoretic syntax to formalize linguistic theories, instead of *being* a linguistic theory, this dependence is clear here. In the case of semantic theory, coverage of linguistic phenomena depends, because of the principle of compositionality, on syntactic representations from which the semantic representations are built.

ACKNOWLEDGEMENTS

We thank Uwe Mönnich, Patrick Blackburn, Reinhard Muskens, Lawrence Stout, and Johan van Benthem for useful comments and encouragement at various points. Hans-Jörg Tiede would like to acknowledge support from Illinois Wesleyan University in the form of an Artistic/Scholarly Development Grant and a Junior Faculty Leave.

BIBLIOGRAPHY

[1] Loredana Afanasiev, Patrick Blackburn, Ioanna Dimitriou, Bertrand Gaiffe, Evan Goris, Maarten Marx, and Maarten de Rijke. PDL for ordered trees. *Journal of Applied Non-Classical Logic*, 15(2):115–135, 2005.

[2] Rolf Backofen, James Rogers, and K. Vijay-Shanker. A first-order axiomatization of the theory of finite trees. *Journal of Logic, Language and Information*, 4(1):5–39, 1995.

[3] Yaniv Bargury and Johann Makowsky. The expressive power of transitive closure and 2-way multihead automata. In Egon Börger, Gerhard Jäger, and Hans Kleine Büning, editors, *Computer science logic (Berne, 1991)*. Springer, Berlin, 1992.

[4] Johan van Benthem and Alice ter Meulen, editors. *Handbook of logic and language*. Elsevier, Amsterdam, 1997.

[5] Robert I. Binnick. The project on annotated bibliography of contemporary research in tense, grammatical aspect, *Aktionsart*, and related areas. `http://www.scar.utoronto.ca/~binnick/TENSE/logic.html`, 2005.

[6] Patrick Blackburn. Tense, temporal reference and tense logic. *Journal of Semantics*, 11:83–101, 1994.

[7] Patrick Blackburn, Claire Gardent, and Wilfried Meyer Viol. Talking about trees. In Steven Krauwer, Michael Moortgat, and Louis des Tombe, editors, *Sixth Conference of the European Chapter of the Association for Computational Linguistics — Proceedings of the Conference*. ACL, 1993.

[8] Patrick Blackburn and Wilfried Meyer-Viol. Linguistics, logic and finite trees. *Bulletin of the Interest Group in Pure and Applied Logics*, 2(1):3–29, 1994.

[9] Patrick Blackburn, Wilfried Meyer-Viol, and Maarten de Rijke. A proof system for finite trees. In Dirk van Dalen and Marc Bezem, editors, *Computer science logic (Paderborn, 1995)*. Springer, Berlin, 1996.

[10] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6:66–92, 1960.

[11] John P. Burgess. Basic tense logic. In Dov Gabbay and Franz Guenther, editors, *Handbook of Philosophical Logic*, volume 2. Reidel, Dordrecht, 1984.

[12] Wojciech Buszkowski. Type logics in grammar. In Vincent F. Hendricks and Jacek Malinowski, editors, *Trends in logic*. Kluwer, Dordrecht, 2003.

[13] John Cantwell. Comparatives. *Theoretical Linguistics*, 21(2-3):145–158, 1995.

[14] Rudolf Carnap. *Meaning and Necessity*. University of Chicago Press, Chicago, second edition, 1956.

[15] Gennaro Chierchia and Sally McConnell-Ginet. *Meaning and Grammar: An Introduction to Semantics*. The MIT Press, Cambridge, second edition, 2000.

[16] Noam Chomsky. Three models for the description of language. *IRE Transactions of information theory*, 2-3:113–124, 1956.

[17] Noam Chomsky. *Syntactic Structures*. Mouton, The Hague, 1957.

[18] Noam Chomsky. *Lectures on Government and Binding*. Foris Publications, Dordrecht, 1981.

[19] Joëlle Cohen, Dominique Perrin, and Jean-Eric Pin. On the expressive power of temporal logic. *Journal of Computer and System Sciences*, 46(3):271–294, June 1993.

[20] Thomas Cornell. Parsing and grammar engineering with tree automata. In Anton Nijholt Dirk Heylen and Giuseppe Scollo, editors, *Algebraic Methods in Language Processing AMiLP 2000, Iowa City, Iowa*, 2000.

[21] Dick Crouch. Temporality in natural language. ESSLLI class notes, 1998.

[22] Henriette de Swart. *Introduction to Natural Language Semantics*. CSLI Publications, Stanford, CA, 1998.

[23] John Doner. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4:406–451, 1970.

[24] David R. Dowty, Robert E. Wall, and Stanley Peters. *Introduction to Montague Semantics*. Reidel, Dordrecht, 1989.

[25] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Springer, Berlin, 1995.

[26] Kousha Etessami and Thomas Wilke. An until hierarchy and other applications of an Ehrenfeucht-Fraïssé game for temporal logic. *Information and Computation*, 160(1/2):88–108, July 2000.

[27] Jeanne Ferrante and Charles W. Rackoff. *The computational complexity of logical theories*. Springer, Berlin, 1979.

[28] Anette Frank. *Context Dependence in Modal Constructions*. PhD thesis, University of Stuttgart, 1997.

[29] Gottlob Frege. On sense and reference. In Peter Geach and Max Black, editors, *Translations from the Philosophical Writings of Gottlob Frege*. Blackwell, Oxford, 1960.

[30] Dov Gabbay. Functional completeness in tense logic. In Uwe Mönnich, editor, *Aspects of Philosophical Logic*. Reidel, Dordrecht, 1981.

[31] Dov Gabbay, Amir Pnueli, Saharon Shelah, and Jonathan Stavi. On the temporal analysis of fairness. In *Proc. 7th Annual ACM Symposium on Principles of Programming Languages*, 1980.

[32] L. T. F. Gamut. *Logic, Language and Meaning*, volume 2. The University of Chicago Press, Chicago, 1991.

[33] Gerald Gazdar, Ewan Klein, Geoffrey Pullum, and Ivan Sag. *Generalized Phrase Structure Grammar*. Harvard University Press, Cambridge, 1985.

[34] Ferenc Gécseg and Magnus Steinby. Tree languages. In *Handbook of formal languages, Vol. 3*. Springer, Berlin, 1997.

[35] Saul Gorn. Explicit definitions and linguistic dominoes. In *Systems and Computer Science (Proc. Conf., London, Ont., 1965)*. Univ. Toronto Press, Toronto, Ont., 1967.

[36] Irene Heim and Angelika Kratzer. *Semantics in Generative Grammar*. Blackwell, Malden, MA, 1998.

[37] Erhard W. Hinrichs. Tense, quantifiers, and contexts. *Computational Linguistics*, 14(2):3–14, 1988.

[38] Jaakko Hintikka and Gabriel Sandu. *On the Methodology of Linguistics*. Basil Blackwell, Oxford, 1991.

[39] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, 1979.

[40] Neil Immerman. *Descriptive Complexity*. Springer, Berlin, 1999.

[41] Theo M. V. Janssen. Compositionality. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*. Elsevier, Amsterdam, 1997.

[42] Hans Kamp. Formal properties of "now". *Theoria*, 37:227–273, 1971.

[43] Hans Kamp and Uwe Reyle. *From Discourse to Logic*. Kluwer, Dordrecht, 1993.

[44] Johan Anthony Willem Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, University of California, Los Angeles, 1968.

[45] Bakhadyr Khoussainov and Anil Nerode. *Automata theory and its applications*. Birkhäuser, Boston, 2001.

[46] Nils Klarlund. Mona & fido: The logic-automaton connection in practice. In Georg Gottlob, Etienne Grandjean, and Katrin Seyr, editors, *Computer Science Logic (Brno, 1998)*, Berlin, 1998. Springer.

[47] Hans-Peter Kolb, Jens Michaelis, Uwe Mönnich, and Frank Morawietz. An operational and denotational approach to non-context-freeness. *Theoretical Computer Science*, 293(2):261–289, 2003.

[48] Andràs Kornai and Geoffrey K. Pullum. The X-bar theory of phrase structure. *Language*, 66:24–50, 1990.

[49] Marcus Kracht. Syntactic codes and grammar refinement. *Journal of Logic, Language and Information*, 4(1):41–60, 1995.

[50] Marcus Kracht. Inessential features. In Alain Lecomte, François Lamarche, and Guy Perrier, editors, *Logical aspects of computational linguistics*. Springer, Berlin, 1997.

[51] Marcus Kracht. On reducing principles to rules. In Patrick Blackburn and Maarten de Rijke, editors, *Specifying Syntactic Structures*. CSLI Publications, Stanford, CA, 1997.

[52] Marcus Kracht. *Tools and techniques in modal logic*. North-Holland, Amsterdam, 1999.

[53] Marcus Kracht. Logic and syntax—a personal perspective. In Michael Zakharyaschev, Krister Segerberg, Maarten de Rijke, and Heinrich Wansing, editors, *Advances in modal logic, Vol. 2*. CSLI Publications, Stanford, CA, 2001.

[54] Marcus Kracht. *The Mathematics of Language*. de Gruyter, Berlin, 2003.

[55] Angelika Kratzer. The notational category of modality. In Hans-Jürgen Eikmeyer and Hannes Rieser, editors, *Words, Worlds, and Contexts*, pages 38–74. de Gruyter, BERLIN, 1981.

[56] Tore Langholm. A descriptive characterisation of indexed grammars. *Grammars*, 4(3):205–262, 2001.

[57] Shalom Lappin, editor. *The Handbook of Contemporary Semantic Theory*. Blackwell Publishers, Oxford, 1996.

[58] Clemens Lautemann, Thomas Schwentick, and Denis Thérien. Logics for context-free languages. In Leszek Pacholski and Jerzy Tiuryn, editors, *Computer science logic (Kazimierz, 1994)*. Springer, Berlin, 1995.

[59] David Lewis. General semantics. In Donald Davidson and Gilbert H. Harman, editors, *Semantics of Natural Language*. Reidel, Dordrecht, 1972.

[60] David Lewis. *Counterfactuals*. Harvard University Press, Cambridge, 1973.

[61] Leonid Libkin. *Elements of finite model theory*. Springer, Berlin, 2004.

[62] Eric Martin and Daniel N. Osherson. *Elements of Scientific Inquiry*. MIT Press, Cambridge, MA, 1998.

[63] Maarten Marx. Conditional XPath, the first order complete XPath dialect. In *Proceedings of PODS '04*, 2004.

[64] James D. McCawley. *Everything that Linguists Have Always Wanted to Know about Logic (But Were Ashamed to Ask)*. University of Chicago Press, Chicago, second edition, 1993.

[65] James D. McCawley. Departmental web site. `http://humanities.uchicago.edu/depts/linguistics/faculty/mccawley.html`, 1998.

[66] Robert McNaughton and Seymour Papert. *Counter-Free Automata*. MIT Press, Cambridge, MA, 1971.

[67] Yiannis N. Moschovakis. Sense and denotation as algorithm and value. In Juha Oikkonen and Jouko Väänänen, editors, *Logic Colloquium '90 (Helsinki, 1990)*. Springer, Berlin, 1993.

[68] Yuko Murakami. *Modal Logic of Partitions*. PhD thesis, Indiana University, Bloomington, IN, 2005.

[69] Reinhard Muskens, Johan van Benthem, and Albert Visser. Dynamics. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*. Elsevier, Amsterdam, 1997.

[70] Adi Palm. Propositional tense logic for finite trees. In *Proceedings of Mathematics of Language (MOL 6)*, 1999.

[71] Adi Palm. Model theoretic syntax and parsing: An application to temporal logic. In *Proceedings of Formal Grammar and Mathematics of Language (FGMOL)*, 2001.

[72] Barbara Partee, Alice ter Meulen, and Robert E. Wall. *Mathematical Methods in Linguistics*. Kluwer, Dordrecht, 1990.

[73] Barbara H. Partee. Some structural analogies between tenses and pronouns. *The Journal of Philosophy*, 70(18):601–609, 1973.

[74] Barbara H. Partee. Possible worlds in model-theoretic semantics: A linguistic perspective. In Sture Allén, editor, *Possible Worlds in Humanities, Arts and Sciences: Proceedings of Nobel Symposium 65*. de Gruyter, Berlin, 1988.

[75] Barbara H. Partee. Montague grammar. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*. Elsevier, Amsterdam, 1997.

[76] John Perry. Possible worlds and subject matter. In Sture Allén, editor, *Possible Worlds in the Humanities, Arts and Sciences, Proceedings of the Nobel Symposium 65*. de Gruyter, Berlin, 1989.

[77] Jean-Eric Pin. Logic, semigroups and automata on words. *Annals of Mathematics and Artificial Intelligence*, 16:343–384, 1996.

[78] Roland Posner. Believing, causing, intending. In René Jorna, Barend van Heusden, and Roland Posner, editors, *Signs, Search, and Communication.* de Gruyter, Berlin, 1992.

[79] Andreas Potthoff. *Logische Klassifizierung regulärer Baumsprachen.* PhD thesis, Christian-Albrechts-Universität zu Kiel, 1994.

[80] Andreas Potthoff. Modulo-counting quantifiers over finite trees. *Theoretical Computer Science*, 126(1):97–112, 1994.

[81] Andreas Potthoff and Wolfgang Thomas. Regular tree languages without unary symbols are star-free. In *Fundamentals of computation theory (Szeged, 1993).* Springer, Berlin, 1993.

[82] Hans Reichenbach. *Elements of Symbolic Logic.* University of California Press, Berkeley, 1947.

[83] James Rogers. *A descriptive approach to language-theoretic complexity.* Studies in Logic, Language and Information. CSLI Publications, Stanford, CA, 1998.

[84] James Rogers. Syntactic structures as multi-dimensional trees. *Research on Language and Computation*, 1(3-4):265–305, 2003.

[85] William C. Rounds. LFP: A logic for linguistics descriptions and an analysis of its complexity. *Computational Linguistics*, 14(4):1–9, 1988.

[86] J. J. M. M. Rutten. Automata and coinduction (an exercise in coalgebra). In Davide Sangiorgi and Robert de Simone, editors, *CONCUR'98: concurrency theory.* Springer, Berlin, 1998.

[87] Bernd-Holger Schlingloff. On the expressive power of modal logics on trees. In Anil Nerode and Michael A. Taitslin, editors, *Logical Foundations of Computer Science - Tver '92.* Springer, Berlin, 1992.

[88] Stuart Shieber. Direct parsing of ID/LP grammars. *Linguistics and Philosophy*, 7:135–154, 1984.

[89] Stuart Shieber. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343, 1985.

[90] Robert C. Stalnaker. A theory of conditionals. In William L. Harper, Robert Stalnaker, and Glenn Pearce, editors, *Ifs.* Reidel, Dordrecht, 1981.

[91] Mark Steedman. *The Productions of Time: Temporality and Causality in Linguistic Semantics.* ms., 2002.

[92] Howard Straubing. *Finite automata, formal logic, and circuit complexity.* Birkhäuser, Boston, 1994.

[93] Howard Straubing, Denis Thérien, and Wolfgang Thomas. Regular languages defined with generalized quantifiers. *Information and Computation*, 118(2):289–301, 1995.

[94] Alice G. B. ter Meulen, editor. *Representing Time in Natural Language: The Dynamic Interpretation of Tense and Aspect.* MIT Press, Cambridge, MA, 1997.

[95] James W. Thatcher. Characterizing derivation trees of context-free grammars through a generalization of finite automata theory. *Journal of Computer and System Sciences*, 1:317–322, 1967.

[96] James W. Thatcher. Tree automata: an informal survey. In *Currents in the theory of computing.* Prentice-Hall, Englewood Cliffs, N. J., 1973.

[97] James W. Thatcher and Jesse B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, 2:57–81, 1968.

[98] Wolfgang Thomas. Languages, automata, and logic. In *Handbook of formal languages, Vol. 3.* Springer, Berlin, 1997.

[99] Richmond H. Thomason, editor. *Formal Philosophy: Selected Papers of Richard Montague.* Yale University Press, New Haven, 1974.

[100] Richmond H. Thomason. Combinations of tense and modality. In Dov Gabbay and Franz Guenthner, editors, *Handbook of Philosophical Logic*, volume 7. Kluwer, Dordrecht, second edition, 2002.

[101] Hans-Jörg Tiede. Proof theory and formal grammars: applications of normalization. In Benedikt Löwe, Wolfgang Malzkom, and Thoralf Räsch, editors, *Foundations of the formal sciences II.* Kluwer, Dordrecht, 2003.

[102] Hans-Jörg Tiede. Inessential features, ineliminable features, and modal logics for model theoretic sytax. In *Proceedings of Formal Grammar and Mathematics of Language (FGMOL)*, 2005.

[103] Johan van Benthem. *The Logic of Time.* Reidel, Dordrecht, 1983.

[104] Johan van Benthem. Temporal logic. In Dov Gabbay, Chris Hogger, and J. Alan Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4. Oxford University Press, Oxford, 1995.

[105] Michiel van Lambalgen and Fritz Hamm. *The Proper Treatment of Events.* Blackwell, Malden, MA, 2005.

[106] Franz von Kutschera. Indicative conditionals. *Theoretical Linguistics*, 1(3):257–269, 1974.

[107] Zsófia Zvolensky. Is a possible-worlds semantics of modality possible? A problem for Kratzer's semantics. In Brendan Jackson, editor, *Proceedings of SALT XII.* Cornell University Press, Ithaca, 2002.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

20

MODAL LOGIC FOR GAMES AND INFORMATION

Wiebe van der Hoek and Marc Pauly

1 Introduction . . . 1078
2 Game Theory . . . 1080
  2.1 Games in Strategic Form . . . 1080
  2.2 Games in Extensive Form . . . 1082
  2.3 Cooperation in Games . . . 1087
3 Game Forms and Kripke Models . . . 1090
  3.1 Players and Outcomes . . . 1091
  3.2 Formalising Solution Concepts in Modal Languages . . . 1092
  3.3 Games as Process Models . . . 1094
  3.4 Other Issues . . . 1096
4 Epistemic Logic . . . 1097
5 Interpreted Systems . . . 1101
6 Games with Imperfect Information . . . 1107
  6.1 Case study: Perfect Recall . . . 1108
  6.2 Outlook . . . 1110
7 Dynamic Epistemic Logic . . . 1110
  7.1 Public Announcements . . . 1111
  7.2 General Updates . . . 1115
8 Epistemic Foundations of Solution Concepts . . . 1121
  8.1 Epistemic Foundations for Strategic Games . . . 1122
  8.2 Epistemic Foundations for Extensive Games . . . 1126
  8.3 On the Representation of Games with Imperfect Information . . . 1129
9 Game Logic . . . 1130
  9.1 Syntax and Semantics . . . 1130
  9.2 Metatheory . . . 1132
  9.3 Other Topics . . . 1134
10 Coalition Logic . . . 1135
  10.1 Syntax and Semantics . . . 1135
  10.2 Axiomatics . . . 1136
  10.3 Complexity . . . 1137
11 Alternating-time Temporal Logic . . . 1138
  11.1 Syntax and Semantics . . . 1138
  11.2 An Example . . . 1140
  11.3 Axiomatics and Complexity . . . 1141
  11.4 Extensions: μ-calculus and imperfect information . . . 1142
12 Conclusion . . . 1143

1 INTRODUCTION

Game-theoretic ideas have a long history in logic. A game-theoretic interpretation of quantification goes back at least to C.S. Peirce, and game-theoretic versions for all essential logical notions (truth in a model, validity, model comparison) have been developed subsequently. The connections between game theory and *modal* logic, on the other hand, have been developed only more recently. At the time of writing, the area is still an active one, so active in fact, that one might even argue that it is too early to be included in a handbook of modal logic. In spite of this concern, we believe that the volume of research falling into the category of modal logic and game theory justifies a survey. Our attempt in this chapter is to put some structure on the various strands of research, to create an organisation which highlights what we consider to be the essential lines of research. As with all endeavours of this sort, one cannot include all the research considered valuable, and we are well aware that our choice of topics also reflects our own interests and expertise.

Game theory has developed a wealth of interesting ideas for describing interactions which may involve a conflict of interest. So far, the logic community has restricted its attention to relatively few of these, mainly studying 2-player extensive games of perfect information which are strictly competitive (even win/lose). In fact, even this restricted class of games has turned out to be extremely rich, as set theory and computer science can testify. Still, given this traditionally narrow focus of logic, it is encouraging to see that more recent work in logic has extended the game-theoretic toolbox considerably, introducing, e.g., cooperative game theory, imperfect information and games involving more than 2 players. While even this can only be a beginning, by now the game-theoretic ideas used in logic certainly go beyond the intuitively natural idea of a winning strategy, and hence we will start our chapter with a section explaining the necessary background in game theory.

Having packed our game-theoretic baggage, our tour starts in Section 3 with new sights of a familiar landscape, possibly the most natural way to link games to modal logic. Game trees can be viewed as Kripke models, where the possible moves are modeled by an accessibility relation and additional information about payoffs and turn taking are encoded by propositional atoms. Structural equivalence notions such as bisimulation then turn into game equivalence notions, and we can investigate extensions of the modal language which can capture game-theoretic solution concepts such as the subgame-perfect equilibrium.

Leaving the first three sections together with the concluding Section 12 aside, the rest of this chapter can be divided into three reading tracks, epistemic logics (Sections 4-8), game logic (Section 9), and coalition logics (Sections 10-11), which can be pursued independently. The epistemic logic track describes approaches based on (dynamic) epistemic logic for dealing with imperfect information in games. The section on game logic describes an extension of Propositional Dynamic Logic for reasoning about games, focusing on operations for combining games like programs. The coalition logic track, finally, discusses a range of logics developed for modeling coalitional power in games, possibly also adding temporal or epistemic operators.

In Section 4 we introduce a widely accepted logic for knowledge in the area of games, where the assumptions impose that players are fully introspective and their knowledge is veridical. The more interesting properties are to be found when studying knowledge of

groups of players though, with *common knowledge* being the main and most intriguing notion in this palette. Section 5 introduces *interpreted systems*, a dominant paradigm in computer science to deal with knowledge and time, where time corresponds with steps in a protocol, or, for our purposes, indeed a game.

Such epistemic notions as introduced in Section 4 and 5 play an important role in games of imperfect information, and, as some major results in early game theory indicate, even beyond that (see Section 8). For instance, a procedure that yields a Nash equilibrium in extensive games, called *backward induction*, finds its justification in the assumption about common knowledge about rationality of the players. We will see, however, that nowadays epistemologists put the need for the inherently infinite conjunctions that come along with common knowledge into perspective and a non-trivial analysis of games can be given without falling back on such strong assumptions.

Where the emphasis in Section 4 is on the knowledge that the players have *about* the game, in Sections 7 and 8 the emphasis will be on how knowledge evolves *during* a game. Hiding one's knowledge can be beneficial for a player within a game, but revealing his ignorance can also be disastrous, and may benefit other players. Moreover, in certain games (like *Cluedo* and many card games) the winning conditions are purely epistemic: the game ends in a win for that player who is the first who happens to know some crucial information.

The dynamic logic of games discussed in Section 9 takes Propositional Dynamic Logic as its starting point. By a change in the underlying semantics, programs become 2-player games which can be combined using the old program operations of sequential composition, test, etc. Besides these program operations, a new duality operator is added which interchanges the roles of the players. Using this new operator, nondeterministic choice splits into two versions depending on which player makes the choice. A typical formula $[(a \cap b); (a \cup c)]p$, for instance, expresses that player 2 has a strategy for achieving p in the game where first, player 1 chooses between a and b, and then player 2 chooses between a and c (for details, see Section 9).

Section 10 introduces Coalition Logic, a basic modal logic for reasoning about the ability of groups in different kinds of games. For a set of individuals C, the formula $[C]p$ expresses that the members of C have a joint strategy for achieving p at the next stage of game. In Section 11, this language is extended to Alternating-time Temporal Logic (ATL) by adding operators for talking about the long-term future, where we can state, e.g., that a coalition can achieve p eventually. ATL is a game-theoretic generalisation of Computation Tree Logic (CTL), with applications in the formal verification of multi-agent systems. Further extensions of ATL to ATL*, the alternating μ-calculus and ATEL are presented. ATEL adds epistemic operators to ATL in order to express, e.g., that a coalition has a strategy for getting an agent to know something eventually.

While our focus in this chapter is on *modal logic for games*, there are also many *games for modal logic*. The reader interested in this reverse connection is referred to other chapters of this handbook (i.e., Chapter 12 and 17) for more details. The similarity between programs and games and the relevance of epistemic and temporal issues, on the other hand, suggest that modal logic may provide an interesting new perspective on games, and it is this perspective we would like to present in this chapter.

2 GAME THEORY

The purpose of this section is to introduce the basic game-theoretic notions needed for the logics discussed in later sections. Hence, this section will also give the reader an indication of the size and nature of the game-theoretic territory which has come under logical investigation.

In Sections 2.1 and 2.2, we discuss games in strategic and extensive forms, respectively. We cover some central solution concepts developed for these models, namely Nash equilibria and subgame-perfect equilibria. For a more detailed discussion of these notions, standard texts on game theory (e.g., [56, 11]) can be consulted. Section 2.3 focuses on a game-theoretic model of cooperation (effectivity functions) which has been investigated in social choice theory [52] and which will play a central role in the logics discussed in Sections 9, 10 and 11.

2.1 Games in Strategic Form

One of the most general models for situations of strategic interaction is that of a strategic game. Because of its generality, strategic games form the standard model in non-cooperative game theory. In a strategic game, the different players choose one of their available alternative actions/strategies, and taken together, these actions determine the outcome of the game. Note that we do not distinguish actions from strategies in strategic games; in extensive games, we will distinguish these two notions. Also, note that game *forms* can be conceived of as 'uninterpreted games': they only deal with the structure of the game, determining which moves are possible in which states, but they do not specify which states are 'good' or 'bad' for any player, i.e., they say nothing about winning, losing, a payoff or utility, when a particular state is reached.

DEFINITION 1 (Strategic Game Form). A *strategic game form* $F = (N, \{\Sigma_i | i \in N\}, o, S)$ consists of a nonempty finite set of agents N, a nonempty set of strategies or actions Σ_i for every player $i \in N$, a nonempty set of states S and an outcome function $o : \Pi_{i \in N}\Sigma_i \to S$ which associates to every tuple of strategies of the players (strategy profile) an outcome state in S. $\dashv$

For notational convenience, let $\sigma_C := (\sigma_i)_{i \in C}$ denote the strategy tuple for coalition $C \subseteq N$ which consists of player i choosing strategy $\sigma_i \in \Sigma_i$. Then given two strategy tuples σ_C and $\sigma_{\overline{C}}$ (where $\overline{C} := N \setminus C$), $o(\sigma_C, \sigma_{\overline{C}})$ denotes the outcome state associated with the strategy profile induced by σ_C and $\sigma_{\overline{C}}$. We shall also write $-i$ for $N \setminus \{i\}$.

Figure 1 below provides an example of a strategic game form among three players in the usual matrix depiction. Unless noted otherwise, we will assume that player 1 chooses the row, player 2 the column, and the third player chooses between the left and the right table. In this example, let σ_1 be the strategy where player 1 chooses B, σ_2 the strategy where player 2 chooses M, and let σ_3 be the strategy of player 3 choosing the left table. Then we have $o(\sigma_{\{1,2\}}, \sigma_{\{3\}}) = o((\sigma_1, \sigma_2, \sigma_3)) = s_1$.

To make a strategic game form into a strategic game, we need to add preference relations or utility functions which express the players' preferences over the game's outcomes. In the first case, given a preference relation $\succeq_i \subseteq S \times S$ for every player $i \in N$ and a strategic game form $F = (N, \{\Sigma_i | i \in N\}, o, S)$, we call $G = (F, (\succeq_i)_{i \in N})$ a *strategic game*. We interpret $s \succeq_i t$ to mean that player i prefers outcome s at least as much as outcome t, and one usually assumes that $\succeq_i$ is a linear order (although this assumption

	L	M	R
T	s_1	s_2	s_1
B	s_2	s_1	s_3

	L	M	R
T	s_3	s_2	s_1
B	s_2	s_3	s_3

Figure 1. A strategic game form for three players.

shall not be essential for our development later). Similarly, given a strategic game form F and a utility function $u_i : S \to \mathbb{R}$ for every player $i \in N$, we can construct a *strategic game* $G = (F, (u_i)_{i\in N})$ where player i prefers outcome s at least as much as outcome t iff $u_i(s) \geq u_i(t)$. Provided that $\succeq_i$ is indeed a linear order, the two definitions are interchangeable, and we shall freely switch between the two formats.

Figure 2 below shows three well-known 2-player games. The matrices list the players' utilities or payoffs, e.g., the top leftmost entry of the leftmost game, $(-4,-4)$ denotes the pair $(u_1(D,D), u_2(D,D))$.

	D	C
D	$-4,-4$	$0,-8$
C	$-8,0$	$-1,-1$

	B	F
B	$2,1$	$0,0$
F	$0,0$	$1,2$

	H	T
H	$-1,1$	$1,-1$
T	$1,-1$	$-1,1$

Figure 2. Three strategic 2-player games: Prisoner's Dilemma (left), Battle of the Sexes (middle) and Matching Pennies (right)

In the Prisoner's Dilemma, two prisoners are interrogated by the police. If the prisoners cooperate (C) and remain silent, they can only be sentenced for a minor offence and will receive one year in prison each. If both defect and confess (D), each will receive 4 years in prison. Finally, if only one prisoner defects, he will go free in order to be used as a witness against his fellow prisoner who will receive 8 years in prison. In the Battle of the Sexes, a couple needs to decide whether to go see a ballet performance (B) or a football match (F) in the evening. Both of them mainly want to spend the evening together, but she prefers the football match and he prefers the ballet performance. Lastly, in the Matching Pennies example, two children each have a penny, and they decide simultaneously whether to show heads (H) or tails (T). One child wins (payoff 1 for the winner, -1 for the loser) in case the sides match, the other child wins in case they differ. Matching Pennies is an example of a *zero-sum* or *strictly competitive* game: For every outcome state s we have that $u_1(s) + u_2(s) = 0$.

A strategic game allows us to model multi-agent interaction using strategies and preferences. Game theory has developed a number of solution concepts which specify a "predicted" set of outcomes for such a game (views differ as to how exactly such a solution has to be interpreted). The following notion is one of the cornerstones of modern game theory.

DEFINITION 2 (Nash Equilibrium). A strategy profile σ_N is a *Nash equilibrium* of a strategic game $G = (N, \{\Sigma_i | i \in N\}, o, S, (\succeq_i)_{i\in N})$ iff $\forall i \in N\ \forall \tau_i \in \Sigma_i :\ o(\sigma_i, \sigma_{-i}) \succeq_i o(\tau_i, \sigma_{-i})$ $\dashv$

Intuitively, a strategy profile (σ_1, σ_2) is a Nash equilibrium in a 2-player game in case σ_1 is a best response to σ_2 and vice versa; no player can improve his payoff by unilaterally

changing his strategy. In the three examples given in Figure 2, the reader may wish to verify that (D, D) is the only Nash equilibrium in the Prisoner's Dilemma, and that both (B, B) and (F, F) are Nash equilibria in the Battle of the Sexes. There is no Nash equilibrium in Matching Pennies.

While for a given player i, a Nash equilibrium only requires i's action to be optimal given the other players' actions, a *dominant* strategy is optimal regardless of what the other players do. Formally, for two strategies $x, y \in \Sigma_i$, x *strictly dominates* y iff $\forall \sigma_{-i} : o(x, \sigma_{-i}) \succ_i o(y, \sigma_{-i})$. We call a strategy strictly dominated iff it is strictly dominated by some other strategy. In the prisoner's dilemma, cooperation is strictly dominated by defection. No dominated strategies exist in the other games mentioned so far.

Our concern in this chapter is mainly with *pure* strategies which are non-probabilistic. In contrast, a *mixed strategy* allows a player to randomise over his set of strategies, playing each strategy with some probability p where $0 \leq p \leq 1$. In the Matching Pennies game, each player may decide to choose Heads with probability $\frac{1}{2}$. This strategy profile $((\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{1}{2}))$ is a Nash equilibrium over mixed strategies, and Nash's celebrated result states that it is no coincidence that Matching Pennies has such an equilibrium.

THEOREM 3 ([53]). *Every strategic game has a Nash equilibrium over mixed strategies.*

The computational complexity of *finding* a mixed strategy Nash equilibrium (from Theorem 3 we are ensured of its existence) for a 2-player strategic game with finite (pure) strategy sets is in NP, but it is presently not known whether the problem is also NP-hard. According to [57], this is one of the most important concrete open questions on the boundary of P today.

2.2 Games in Extensive Form

Strategic games consider strategic interaction as involving only a single choice for every player. There may be situations, however, where we want to model the fine structure of strategic interaction which involves modelling the sequential structure of decision making. Extensive form games provide us with this level of added detail.

Perfect Information

Given a finite or infinite sequence of actions $h = (a_1, a_2, \ldots)$, let $h|k = (a_1, a_2, \ldots, a_k)$ denote the initial subsequence of length k of h.

DEFINITION 4 (Extensive Game Form of Perfect Information). An *extensive game form of perfect information* is a triple $F = (N, H, P)$, where as before, N is the set of players.

H is a set of sequences (finite or infinite) over a set A of actions which we shall call *histories* (or: *runs*, *plays*) of the game. We require that H satisfies three requirements: (1) the empty sequence $() \in H$. (2) H is closed under initial subsequences, i.e., if $h \in H$ has length l, then for all $k < l$ we have $h|k \in H$. If $h \in H$ is infinite, $h|k \in H$ for all k. (3) If all finite initial subsequences of an infinite sequence h are in H, then so is h: given an infinite sequence h such that for all k we have $h|k \in H$, then $h \in H$. Let $Z \subseteq H$ be the set of *terminal histories*, i.e. $h \in Z$ iff for all $h' \in H$ and k such that $h'|k = h$ we have $h' = h$ (so, infinite histories are terminal). $P : H \backslash Z \to N$ is the player function which assigns to every nonterminal history the player whose turn it is to move. $\dashv$

As with strategic games, we turn an extensive game form into an extensive game

by adding preference relations. Formally, let $\succeq_i \subseteq Z \times Z$ be a preference relation on the set of terminal histories Z. As before, we shall sometimes use utility functions u_i instead of preference relations. Given an extensive game form $F = (N, H, P)$, we call $G = (F, (\succeq_i)_{i \in N}) = (N, H, P, (\succeq_i)_{i \in N})$ an *extensive game*. We call an extensive game G *finite* iff its set of histories H is finite. G has a *finite horizon* iff all histories in H have finite length.

Figure 3 shows the tree representation of an extensive game, where the branches are labelled by actions and the payoffs of the players are shown at the terminal nodes.

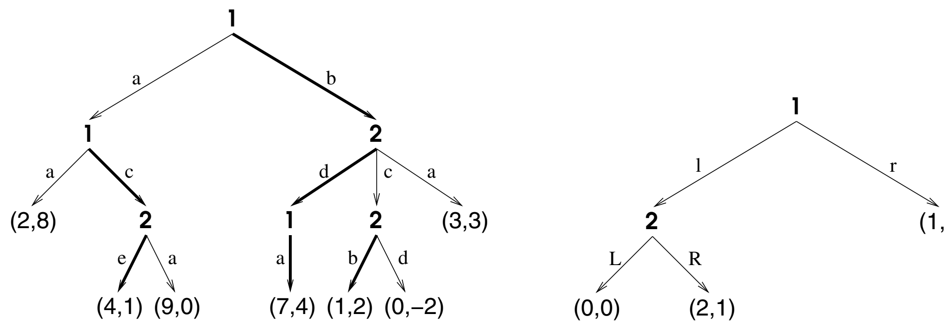


Figure 3. An extensive game for two player with payoffs (left) and an extensive game with an implausible Nash equilibrium (right)

Given a finite history $h = (a_1, \ldots, a_n)$ and an action $x \in A$, let $(h, x) = (a_1, \ldots, a_n, x)$. Furthermore, let $A(h) = \{x \in A | (h, x) \in H\}$ be the set of actions possible after h. Now we can define a strategy for player i as a function $\sigma_i : P^{-1}[\{i\}] \to A$ such that $\sigma_i(h) \in A(h)$ (P^{-1} denotes the pre-image of P). As before, we let Σ_i denote the set of strategies of player i. Given a strategy profile $\sigma = (\sigma_i)_{i \in N}$, let $o(\sigma) \in H$ be the history which results when the players use their respective strategies.

In the game on the left of Figure 3, the strategy for player 1 indicated by bold arrows is given by $\sigma_1(()) = b$, $\sigma_1((a)) = c$ and $\sigma_1((b, d)) = a$. Player 2's strategy indicated in the game is given by $\sigma_2((a, c)) = e$, $\sigma_2((b)) = d$ and $\sigma_2((b, c)) = b$.

The notion of a Nash equilibrium can now be lifted easily from strategic games to extensive games. Given an extensive game $G = (N, H, P, (\succeq_i)_{i \in N})$, a strategy profile σ is a *Nash equilibrium* iff $\forall i \in N\ \forall \tau_i \in \Sigma_i : \quad o(\sigma_i, \sigma_{-i}) \succeq_i o(\tau_i, \sigma_{-i})$. However, it turns out that for extensive games, Nash equilibria often lack plausibility, as shown by the game on the right in Figure 3, taken from [56].

The Nash equilibria of the game are (l, R) and (r, L). The second equilibrium however does not seem reasonable. At the position where player 2 has to move, she will choose R since this will give her a higher payoff. Knowing this, player 1 should choose l at the beginning of the game, and so we would want to advocate only (l, R) as the solution of the game. The strategy profile (r, L) turns out to be a Nash equilibrium because of the threat that player 2 will choose L rather than R, but this threat is not credible since choosing L would hurt her own interest. To rule out such pathological equilibria, we need to strengthen our equilibrium notion. The problem with profile (r, L) is that it prescribes an unreasonable choice in a subgame of the original game, whereas we would want our

equilibrium strategies to be optimal in every subgame.

To obtain a more robust equilibrium notion, we introduce the notion of a subgame more formally. Given two sequences h and h', with h finite, let (h, h') denote the concatenation of h and h'. Consider a history h in the extensive game $G = (N, H, P, (\succeq_i)_{i \in N})$. To isolate the subgame starting after h, we define $G(h) = (N, H|_h, P|_h, (\succeq_i|_h)_{i \in N})$ where $H|_h = \{h' | (h, h') \in H\}$, $P|_h(h') = P(h, h')$ for each $h' \in H|_h$ and finally $h' \succeq_i |_h h''$ iff $(h, h') \succeq_i (h, h'')$. Similarly, strategies σ_i (and strategy profiles) can be restricted to subgames by setting $\sigma_i|_h(h') = \sigma_i(h, h')$.

DEFINITION 5 (Subgame-Perfect Equilibrium). A strategy profile σ is a *subgame-perfect Nash equilibrium* of a game G iff for every history $h \in H$, the restriction $\sigma|_h$ of σ is a Nash equilibrium of $G(h)$. $\dashv$

In the game of Figure 3 (right), only (l, R) is a subgame-perfect equilibrium. Note that all subgame-perfect equilibria are also Nash equilibria. A game is called *generic* if no player is indifferent between any two terminal histories, i.e., for all $i \in N$ and $h, h' \in H$, we have $u_i(h) = u_i(h')$ iff $h = h'$.

THEOREM 6 ([42]). *Every finite extensive game of perfect information has a subgame-perfect equilibrium. Moreover, in any generic game, this equilibrium is unique.*

While the formal proof of this theorem is somewhat technical, the general method used to establish the result is easy to explain and is known as *backward induction*. We build up the equilibrium profile by induction on the length of a game (i.e., the length of its longest history). If a game has length 0, it only consists of a terminal node and there are no strategic decisions to be made. Consider the payoff vector of the terminal node as the backward induction vector (short: bi-vector) of the game. Now if the game G has length $n+1$, assume that player i has to move at the root of G. By induction hypothesis, all the proper subgames of G have a bi-vector and we have associated strategy profiles for them. To get an equilibrium profile for G, let i choose a successor with the highest bi-vector for i, and consider that payoff vector as the bi-vector of G. Starting at the terminal nodes, this backward induction method moves up through the game tree and inductively defines a strategy profile which turns out to be a subgame-perfect equilibrium and a bi-vector which is its associated payoff vector.

Note that the intuitive reasoning which we used to argue against (r, L) as a solution of the game in Figure 3 (right) was already an example of backward induction reasoning. In Figure 3, the backward induction profile in the game to the left has been indicated by boldface arrows. The payoff vector of the backward induction profile is $(7, 4)$.

As a corollary to Theorem 6, we obtain the following well-known result due to Zermelo which (in a slightly generalised version) can be used to show, e.g., that in the game of chess, either black or white must have a strategy which guarantees at least a draw. We say that a 2-player extensive game of perfect information is a *win-loss game* provided that it is strictly competitive and for all histories h, either $u_1(h) = 1$ or $u_2(h) = 1$, i.e., win and loss are the only two possible outcomes. In such a game, a strategy is a *winning strategy* for player i provided it guarantees a history h such that $u_i(h) = 1$.

THEOREM 7 ([107]). *Every finite 2-player win-loss game is determined, i.e., one of the players has a winning strategy.*

One can even show that the problem of determining whether such a game is a win for a specific player can be determined in polynomial time: in fact, it is a ‘canonical P-complete

problem' in the sense that it is a popular candidate to be related to other computational problems (like AND/OR Graph Solvability) in order to show *their* polynomial complexity. See [27, Appendix 11.1].

Almost Perfect Information

Extensive games of perfect information impose a strict order on the moves which take place in a game. As a first generalisation, we may extend the definition of an extensive game to allow for simultaneous moves of the players. These *extensive games of almost-perfect information* (or *extensive games with simultaneous moves*) will be important for our discussion of Coalition Logic and ATL. In these games, players are completely informed about the past, but they may be unsure about the present, i.e., about the actions the other players are simultaneously taking.

Formally, an extensive game of almost-perfect information is a tuple $G = (N, H, P, (\succeq_i)_{i\in N})$ just like an extensive game of perfect information, with the only difference that for every nonterminal history $h \in H$, $P(h)$ is a nonempty subset of N. Furthermore, for each $i \in P(h)$, we have a set $A_i(h)$ of actions possible for player i at h, and we define the set of actions possible after h to be $A(h) = \prod_{i\in P(h)} A_i(h)$. Histories of the game are now sequences of vectors, consisting of the actions chosen simultaneously by the appropriate players. A strategy for player i is now a function σ_i such that $\sigma_i(h) \in A_i(h)$. The definitions of Nash equilibrium and subgame-perfect equilibrium can easily be adjusted to these extensive games with simultaneous moves.

Imperfect Information

So far, we have assumed that the players always know where they are in the game tree. This amounts to assuming that the players are always informed about the actions which have been taken so far, both by the other players and by themselves; in short, we have considered perfect information games where a player has no private information (e.g., cards which only she knows), nor does she ever forget which moves she has made earlier (this is the essence of the game "Memory"). The game model we shall introduce now is an extension of the extensive game model to cover situations of imperfect information.

DEFINITION 8 (Extensive Game of Imperfect Information). An *extensive game of imperfect information* is a tuple $G = (N, H, P, (\mathcal{I}_i)_{i\in N}, (\succeq_i)_{i\in N})$. The only new component $\mathcal{I}_i$ is a partition of the set of histories where i has to move, i.e. of $P^{-1}[\{i\}]$, with the property that for all $h, h' \in I \in \mathcal{I}_i$, $A(h) = A(h')$. ⊣

The elements of $\mathcal{I}_i$ are called *information sets*. If player i has to make a decision in a game at history $h \in I \in \mathcal{I}_i$, she does not know which of the histories in I is the real history, i.e. she considers all histories in I as possible alternatives to h. In order for this interpretation to make sense, we have to assume that the histories in I cannot be distinguished by what actions are possible in the various histories, a requirement which we enforced by demanding that for all $h, h' \in I$, $A(h) = A(h')$. Observe that if all information sets are singletons, we have in fact a game of perfect information.

Since a player cannot distinguish between two histories which are in the same information set, her strategies have to be uniform within every information set. Hence, we define a strategy for an extensive game of imperfect information to be a function $\sigma_i : \mathcal{I}_i \to A$ such that $\sigma_i(I) \in A(I)$. In words, a strategy picks an action for every information set;

since histories within one information set allow for the same actions to be taken, the action prescribed by the strategy can always be executed, no matter where the player really is in the game tree.

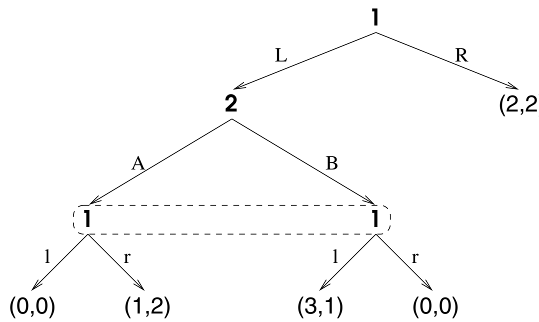


Figure 4. Example of an extensive game of imperfect information

Figure 4 contains an example of an imperfect information game. The information sets which are not singletons have been indicated by drawing a dashed box around the histories which are in the same information set. So in this game, player 1 makes the first move, and assuming she chooses to play L, player 2 moves afterwards. Player 1 however obtains no information about the choice made by player 2, maybe she was not present when the choice was made, maybe she forgot, etc. Her strategy therefore would have to specify either l or r for both cases, since she is unable to distinguish them. Two more examples of imperfect information games are given in Figure 5. The left game exhibits

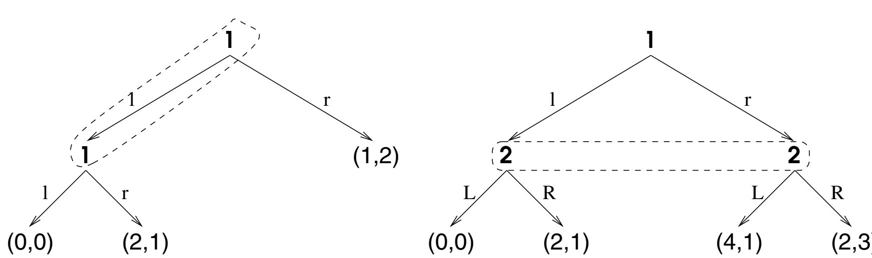


Figure 5. A game with imperfect recall (left) and one with simultaneous moves (right)

imperfect recall: After doing l, player 1 does not know whether she has done l already or not. Since this notion will play a role later, we will define it more formally here. Given game $G = (N, H, P, (\mathcal{I}_i)_{i\in N}, (\succeq_i)_{i\in N})$ and history $h \in H$, let $X_i(h)$ record the experience of player i along history h, i.e., the sequence of information sets the player encounters in h and the actions he takes at that information set. So with the game of Figure 4, we have $X_1(LAr) = (\emptyset, L, \{LA, LB\}, r)$. Then we say that G has *perfect recall* iff for every player i we have $X_i(h) = X_i(h')$ whenever there is some $I \in \mathcal{I}_i$ such that $h, h' \in I$. So while Figure 4 presents a game with perfect recall, the left game of Figure 5 does not, since $X_1(\emptyset) = (\emptyset)$ but $X_1(l) = (\emptyset, l)$.

The right game in Figure 5 shows that the imperfect information game model can also be used to model games with simultaneous moves. After player 1 moves, player 2 has to move without any information about the choice made by the first player. Thus, since player 1 also does not know about the decision that player 2 will make later, we can interpret this game as one where the two players simultaneously choose an action.

The concepts of Nash equilibrium and mixed strategy Nash equilibrium can easily be extended to imperfect information games. The story is more complicated for the subgame-perfect equilibrium. The game in Figure 6 demonstrates the problem that one can run into when wanting to apply backward induction to a game of imperfect information.

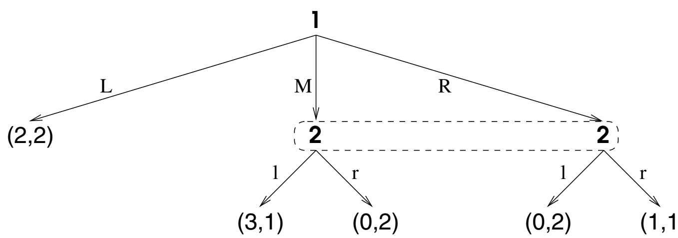


Figure 6. An imperfect information game where backward induction runs into problems

Player 2's only information set contains two subgames. In the left subgame, r is the strategy prescribed by backward induction, whereas in the right subgame, l is optimal. Since both subgames lie in the same information set, the backward induction strategy has to be uniform for both subgames. In order to label one of the two strategies as optimal, player 2 would have to know where she is in the game, but this is exactly what she does not know.

To deal with this problem, one can introduce a belief system which specifies at each information set the probability with which the player believes that a history has happened. The strategy choice can then make use of these probabilities. This leads to the notion of a *sequential equilibrium* which we shall not define formally here. Note simply that for the game in Figure 6, if player 2 believes that it is more likely that player 1 will play M rather than R, strategy r should be preferable to player 2. We refer the reader to [56] for the details.

2.3 Cooperation in Games

So far we have assumed that agents determine individually what strategy they want to follow. We made no attempt to account for the possibility that agents might cooperate in bringing about a desirable state of affairs. Effectivity functions, the model we discuss in this subsection, aim at capturing explicitly the powers which agents can obtain by forming coalitions.

Effectivity functions model the power distribution among individuals and groups of individuals. In social choice theory, they have been used in particular to model voting procedures. The exposition we will give here focuses on providing the necessary background to understand the link between effectivity functions and the neighbourhood

models used in non-normal modal logics like the ones we will discuss in Sections 9, 10 and 11.

Effectivity functions have been studied extensively in game theory and social choice theory [52, 1, 66]. The following exposition is based on [63, 62].

DEFINITION 9 (Effectivity Function). Given the finite nonempty set of players N and a nonempty set of alternatives or states S, an *effectivity function* is any function $E : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ which satisfies the following two conditions: (1) $\forall C \subseteq N : \emptyset \notin E(C)$, and (2) $\forall C \subseteq N : S \in E(C)$. ⊣

The function E associates to every group of players the sets of outcomes for which the group is effective; coalition C is effective for X if it can bring about an alternative in X, even though it may have no control over which alternative of X is realised. The literature differs somewhat in the conditions placed on the function E. The conditions chosen here aim at Theorem 12 and the logics discussed in Sections 10 and 11. Informally, condition (1) of Definition 9 states that no group C can ensure that *nothing* is brought about, while condition (2) expresses that every group C can at least bring about *something* by 'choosing' the complete set of alternatives S, putting no constraints on what the players outside C can achieve.

EXAMPLE 10. Consider the following example from [24] about Angelina, Edwin and the judge: If Angelina does not want to remain single, she can decide either to marry Edwin or the judge. Edwin and the judge each can similarly decide whether they want to stay single or marry Angelina. If we assume that the three individuals live in a society where nobody can be forced to marry against his/her will, this situation can be modelled using effectivity functions as follows: The set of players is $N = \{a, e, j\}$ and the set of alternatives is $S = \{s_s, s_e, s_j\}$, where s_s denotes the situation where Angelina remains single, s_e where she marries Edwin, and s_j where she marries the judge. Angelina (a) has the right to remain single, so $\{s_s\} \in E(\{a\})$, whereas Edwin can only guarantee that he does not marry Angelina; whether she marries the judge or remains single is not up to him. Consequently, we have $\{s_s, s_j\} \in E(\{e\})$ and there is no proper subset X of $\{s_s, s_j\}$ such that $X \in E(\{e\})$. Analogously for the judge, we have $\{s_s, s_e\} \in E(\{j\})$. Angelina and Edwin together can achieve any situation except the one where Angelina marries the judge (since this alternative would require the judge's consent), and hence $\{s_s\}, \{s_e\} \in E(\{a, e\})$. Again, the situation is similar for the judge: $\{s_s\}, \{s_j\} \in E(\{a, j\})$. ⊣

In most situations, coalitional effectivity functions will satisfy some additional properties. Among the central properties are the following:

MONOTONICITY: Since a superset of states places fewer constraints on a coalition's ability, we can usually assume that effectivity functions are *monotonic*: For every coalition $C \subseteq N$, if $X \subseteq X' \subseteq S$, $X \in E(C)$ implies $X' \in E(C)$.

MAXIMALITY: An effectivity function E is *C-maximal* if for all X, if $\overline{X} \notin E(\overline{C})$ then $X \in E(C)$. E is *maximal* iff for all coalitions C it is C-maximal. Instantiating this condition for 2 players over $S = \{win_1, win_2\}$, $\{1\}$-maximality expresses that the game is determined: if one player does not have a winning strategy, then the other player does.

SUPERADDITIVITY: The most interesting principle governs the formation of coalitions. It states that coalitions can combine their strategies to (possibly) achieve more: E is *superadditive* if for all X_1, X_2, C_1, C_2 such that $C_1 \cap C_2 = \emptyset$, $X_1 \in E(C_1)$ and $X_2 \in E(C_2)$ imply that $X_1 \cap X_2 \in E(C_1 \cup C_2)$.

Given utility functions $(u_i)_{i \in N}$ for the players, effectivity functions also allow us to define solution concepts. Given an effectivity function $E : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$, call an alternative $s \in S$ *dominated* if there is a set $X \subseteq S$ and a coalition C such that $X \in E(C)$ and for all $i \in C$ and $x \in X$ we have $u_i(x) > u_i(s)$.

DEFINITION 11 (Core). Given an effectivity function $E : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ and utility functions $(u_i)_{i \in N}$, the *core* of $(E, (u_i)_{i \in N})$ is the set of undominated alternatives. $\dashv$

In connection with the core, it is usually also assumed that for all sets $X \neq \emptyset$, we have $X \in E(N)$, so in particular, any state can be achieved by the grand coalition of all players.

An effectivity function E is *stable* if for any set of utility functions $(u_i)_{i \in N}$, the core of $(E, (u_i)_{i \in N})$ is nonempty. Given E and $(u_i)_{i \in N}$, one can determine in polynomial time whether the core of $(E, (u_i)_{i \in N})$ is nonempty. Determining whether E is stable, however, is an NP-complete problem [50].

From Strategic Games to Effectivity Functions

Effectivity functions can be derived from a strategic game form in a number of different ways. Given a strategic game form G, a coalition $C \subseteq N$ will be *α-effective* for a set $X \subseteq S$ iff the coalition has a joint strategy which will result in an outcome in X no matter what strategies the other players choose. Formally, for a strategic game form $G = (N, (\Sigma_i)_{i \in N}, o, S)$, its *$\alpha$-effectivity function* $E^\alpha_G : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ is defined as follows:

$$X \in E^\alpha_G(C) \text{ iff } \exists \sigma_C \forall \sigma_{\overline{C}} \;\; o(\sigma_C, \sigma_{\overline{C}}) \in X.$$

We say that an effectivity function $E : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ *α-corresponds* to a strategic game G iff $E = E^\alpha_G$.

Analogously, a coalition $C \subseteq N$ will be *β-effective* for a set $X \subseteq S$ iff for every joint strategy of the other players, the coalition has a joint strategy which will result in an outcome in X. Hence, in contrast to α-effectivity, the coalition's strategy may depend on the strategy of the other players. Formally, the *β-effectivity function* $E^\beta_G : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ of a game form G is defined as follows

$$X \in E^\beta_G(C) \text{ iff } \forall \sigma_{\overline{C}} \exists \sigma_C \; o(\sigma_C, \sigma_{\overline{C}}) \in X.$$

It is easy to see that $E^\alpha_G \subseteq E^\beta_G$, i.e. α-effectivity implies β-effectivity, but the converse does not hold, as the example in Figure 7 illustrates. In that game G, player 1 chooses the row, player 2 the column, and the third player chooses between the left and the right table. For every joint strategy of players 1 and 3, player 2 has a strategy which yields outcome s_2. Note, however, that this strategy depends on the strategies chosen by players 1 and 3, i.e., player 2 has no strategy which will guarantee outcome s_2 independent of the strategies of players 1 and 3. In terms of α- and β-effectivity, we have $\{s_2\} \in E^\beta_G(\{2\})$, but $\{s_2\} \notin E^\alpha_G(\{2\})$. The coalition consisting of players 1 and 2 on the other hand does

	l	m	r
l	s_1	s_2	s_1
r	s_2	s_1	s_3

	l	m	r
l	s_3	s_1	s_2
r	s_2	s_3	s_3

Figure 7. A strategic game where α- and β-effectivity differ

have a joint strategy (r, l) which guarantees s_2 independent of player 3's strategy, i.e. $\{s_2\} \in E^{\alpha}_{G}(\{1, 2\})$.

While this discussion shows that every strategic game form can be linked to an effectivity function via α-correspondence, not every effectivity function will be the α-effectivity function of a strategic game form. The properties required to obtain a precise characterisation result are the following.

THEOREM 12 ([63]). *An effectivity function $\mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ α-corresponds to a strategic game form if and only if it is monotonic, N-maximal and superadditive.*

3 GAME FORMS AND KRIPKE MODELS

Figure 3 invites the modal logician to apply his tools and analysis as provided in especially Chapter 1 of this handbook in a natural way: games in extensive form are just models over time or processes in disguise [85]. In other words, if one is interested in the moves and their outcomes, extensive games can be conceived of as Kripke models for dynamic logic: see also Section 9. We take this perspective in this section, swiftly moving the focus of our analysis from the area of games to that of logic. To start with, we abstract from the specific actions, and reason about what specific agents can achieve. We do this in an example driven, and semantic way, in Section 3.1. When combining the players' powers with their preferences, modal logic can help to shed light on solution concepts, an exercise we will undertake in Section 3.2. In these two subsections, it is not the extensive form of the game that matters for the modal analysis so much, but more which end-nodes can be reached. We make some remarks concerning equivalence and expressivity taking the full structure of the extensive game into account, in Section 3.3. Yet another point of view takes the strategies, or, rather, the paths in an extensive game as first class citizens: see Section 11.

Games in Strategic Form

However, before exploiting the modal structure of games in extensive form, let us follow the structure of Section 2.1 and point out that the applicability of modal logic also extends to games in strategic form. The contribution [13] introduces a dynamic relation R_i for every player i, with intended meaning that sR_it holds iff from state s, player i can unilaterally bring about state t. States are states of the game, and σ assigns to every state the strategy profile that is being played in any state: e.g., $\sigma_i(s) = s_i$ denotes that i plays strategy s_i in s. Regarding the atomic propositions, [13] assumes to have atoms $q \leq p$ for any natural numbers p and q (with straightforward interpretation), and, for every player i and number p, an atom $u_i = p$ denoting i's utility (or pay-off) at any state: such an atom is true at s iff $u_i(\sigma(s)) = p$, where u_i assigns a utility $p \in \mathbb{N}$ to

every strategy profile σ. We assume to only need finitely many values $Val = \{p, q, \dots\}$. Finally, the atom $Nash$ is true at all the profiles at which this equilibrium is played.

For instance, using a suggestive name for states, in our representation of the Prisoner's Dilemma (see Figure 2), in state (D, D) it is true that $u_1 = -4 \wedge u_2 = -4$ and, everywhere in the game, we have

$$\Box_2 \Diamond_1 \Big(\bigvee_{q \in Val} (u_1 = q) \wedge (q \geq -4) \Big) \wedge \Diamond_2 \Box_1 \Big(\bigwedge_{p \in Val} (u_1 = p) \rightarrow (p \leq -4) \Big) \tag{1}$$

Equation (1) expresses that, no matter what player 2 does, player one can guarantee himself a payoff of -4, but at the same time, he can do not better than that: 2 can play a move such that the best that 1 can achieve is -4. Note that we *do* have, in the game of Figure 2, that $\Diamond_1 \Diamond_2 ((u_1 > -4) \wedge (u_2 > -4))$ (if the players were to cooperate, they could achieve more than -4 each). Let $p_1, \dots, p_n$ be a set of n payoff values. The following is valid in every strategic game G:

$$\bigwedge_{i \in N} \left((u_i = p_i) \wedge \Box_i \bigwedge_{q \in Val} (u_i = q \rightarrow (q \leq p_i)) \right) \leftrightarrow \left(\bigwedge_{i \in N} (u_i = p_i) \wedge Nash \right) \tag{2}$$

This formula is equivalent to saying that, given that every player i's outcome is p_i, the strategy played is a Nash-outcome if and only if no player i can unilaterily deviate and achieve something (q) that is better than p_i.

3.1 Players and Outcomes

Given an extensive game form $F = (N, H, P)$, we straightforwardly associate with it a *game frame* $\mathcal{F}_F^N = \langle W, R_{i \in N} \rangle$ with the obvious addition that $H = W$ and $R_i st$ iff $P(s) = i$ & $t = (s, a)$ for some action a, i.e., if player i is to move in s and he can choose a move that leads to t. Such a frame is *generated* from a root and, moreover, *turn-based*: if $R_i st$ then for no other j and for no u also $R_j su$. Basic propositions like $\mathbf{turn_i}$ (player i is to move) and **end** (we are in a leaf) can easily be defined, as $\Diamond_i \top$ and $\bigwedge_{i \in N} \Box_i \bot$, respectively. We can also start from an extensive game G and then obtain $\mathcal{F}_G^N$ by augmenting the frame with a preference relation. Moreover, we can assume to have atoms $u_i = p_i$ or $\mathbf{win_i}$ in the language, and interpret them in an appropriate manner in a model $\mathcal{M}_F^N$ or $\mathcal{M}_G^N$. For instance, in the game on the right of Figure 3, we have in the root that $\Diamond_1 (\Box_2 (u_2 \leq 2) \wedge \Diamond_2 \top) \wedge \Diamond_1 \Box_2 \bot$: player 1 can enforce a state such that player 2 can move but is unable to obtain more than 2 units, but player 1 can also move to a state in which player 2 cannot make any move anymore.

The paper [13] argues that modal logic not only is a useful tool to *describe* the rational behaviour of players (see Section 3.2), but also when it comes to *prescribing* the players how to act. To do so, [13] adds a relation R_* to a game frame $\mathcal{F}^N$ representing (paths according to) a *recommendation* that the players are given, i.e., $R_* \subseteq R^N$, where R^N is the transitive closure of $N^T = (\bigcup_{i \in N} R_i)$ with the following properties: (1) R_* is transitive, (if it is recommended to reach w_2 from w_1 and w_3 from w_2, then it is recommended to reach w_3, in w_1); (2) if for some w^T, $N^T w w^T$ (i.e., w is a decision node $\in T = H \setminus Z$), then also $R_* w w_*$ for some w_* (if a player is to move at w, then a recommendation must be made); and, finally (3) if $R_* w_1 w_3$ and at the same time

$R^N w_1 w_2$ and $R^N w_2 w_3$, then we have both $R_* w_1 w_2$ and $R_* w_2 w_3$ (if it is recommended in w_1 to reach w_3, then any path to do that is a recommended path).

Moreover, [13] allows for atomic propositions $(q \leq p)$ and $u_i = p_i$, the latter only being true in a state s iff s is a leaf, with $u_i(s) = p_i$. Now consider the following scheme:

$$\Diamond_*(u_i = p_i) \rightarrow \Box_i\left(\left(\left(u_i = q_i\right) \vee \Diamond_*(u_i = q_i)\right) \rightarrow q_i \leq p_i\right) \tag{3}$$

[13] refers to (3) as *internal consistency* of a recommendation, "in the sense that no player can increase his payoff by deviating from the recommendation, *using the recommendation itself* to *predict his future payoff after the deviation*" ([13, page 17]).

Of course, it is up to the game theoretician to come up with the 'right recommendation', but an obvious choice would be that they play a Nash equilibrium. In a generic game (cf Theorem 13), the backward induction algorithm determines for every decision node a unique immediate successor: let us call this the backward induction relation BI. We say that a recommendation relation R_* is the backward induction recommendation if it is the transitive closure of BI. The next Proposition tells us that scheme (3) can be understood as characterising backward induction.

THEOREM 13. *[13, Proposition 4.8] Let G be a generic perfect information game and $\mathcal{F}_G^N$ its associated Kripke model, with a recommendation relation R_*. Then the following are equivalent:*

(1) R_* is the backward induction recommendation

(2) scheme (3) is valid in $\mathcal{F}_G^N$

3.2 Formalising Solution Concepts in Modal Languages

The aim of [35] is also to formalise solution concepts in a modal logic. We again start with frames $\mathcal{F}_G$ based on a game G with preference relations $\succeq_{i_{i\in N}}$. It assumes that every $\succeq_i$ is reflexive, transitive and connected (i.e., for all u, v, $u \succeq_i v$ or $v \succeq_i u$). Every $\succeq_i$ gives rise to an operator $[i]$, with intuitive reading of $[i]\varphi$: "φ holds in all states at least as preferable to the present one".

The other first class citizens in [35] are *strategy profiles* in the game G. For any such profile σ, let $R_\sigma st$ iff following σ in s would eventually lead to the end state t. Thus, $[\sigma]\varphi$ reads "if from here all players adhere to σ, the play will eventually end in a state in which φ holds". Finally, for every player i and strategy profile σ, recall that (τ_i, σ_{-i}) is the profile where all players stick to σ, except for i who deviates and plays τ_i. We use this to define a third accessibility relation: $R_{(i,\sigma)}st$ iff for some τ_i, $R(\tau_i, \sigma_{-i})st$. Hence, the meaning of $[i,\sigma]\varphi$ in s becomes that "φ holds in all the states that will be reached if all the players except possibly i play the strategy σ". Given an extensive game $G = (N, H, P, (\succeq_i)_{i\in N})$ we now define $\mathcal{F}_G$ as $\langle H, (\succeq_i), (R_\sigma), R_{(i,\sigma)}\rangle$, where $i \in N$ ranges over the players, and σ over the strategy profiles. Note that the binary relations R_σ, and $R_{(i,\sigma)}$ on such a frame have the leaves as their co-domain (cf. Figure 8). By decorating such a frame with a valuation $\pi : H \rightarrow 2^A$ for some set of atoms A, we obtain a *game model* M_G for G.

Now, before stating a result about these relations, we first recall a result from *correspondence theory* (see Chapter 5).

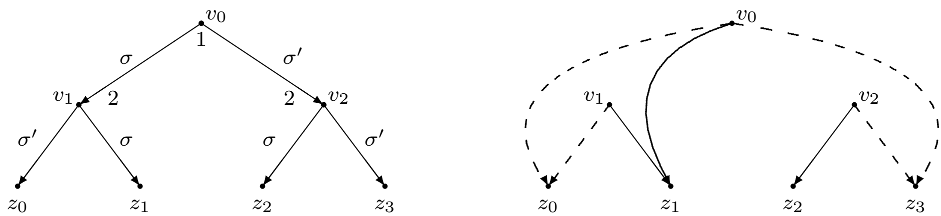


Figure 8. Transformation of an extensive game (left) to a game-frame (right) with respect to two strategy profiles σ and σ' and their corresponding accessibility relations R_σ and $R_{(1,\sigma')}$ (dashed). Reflexive arrows at the leaves are omitted.

THEOREM 14. *Suppose we have three accessibility relations R_k, R_l and R_m with corresponding modalities. Then the scheme $\langle k\rangle[l]\varphi \to [m]\varphi$ characterises frames satisfying (k,l,m)-Euclidicity, i.e., frames in which $\forall s,t,u((R_k st \;\&\; R_m su) \Rightarrow R_l tu)$.*

Remember that a Nash-equilibrium in an extensive game is a strategic profile σ such that for all players i and profiles τ, we have $o(\sigma_i,\sigma_{-i}) \succeq_i o(\tau_i,\sigma_{-i})$, i.e. no player can improve his situation by unilaterally deviating from σ. The profile σ is said to be a *best response* for player i iff for all profiles τ, we have $o(\sigma_i,\sigma_{-i}) \succeq_i o(\tau_i,\sigma_{-i})$. This is an individual pendant of Nash-equilibrium: clearly, a strategy profile σ is a Nash-equilibrium iff it is a best response for all players.

THEOREM 15 ([35], Theorem 3.1). *Let $\mathcal{F}_G$ be obtained from the extensive game G as indicated above, and let v_0 be its root-node. Let 's.p.' stand for 'sub-game perfect'.*

$$\begin{array}{lrl}
(i) & \sigma \text{ is a best response for } i \text{ in } G \text{ iff} & \mathcal{F}_G, v_0 \models \langle i,\sigma\rangle[i]\varphi \to [\sigma]\varphi \\
(ii) & \sigma \text{ is an s.p. best response for } i \text{ in } G \text{ iff} & \mathcal{F}_G \models \langle i,\sigma\rangle[i]\varphi \to [\sigma]\varphi \\
(iii) & \sigma \text{ is a Nash equilibrium in } G \text{ iff} & \mathcal{F}_G, v_0 \models \bigwedge_{i\in N}\left(\langle i,\sigma\rangle[i]\varphi \to [\sigma]\varphi\right) \\
(iv) & \sigma \text{ is an s.p. (Nash) equilibrium in } G \text{ iff} & \mathcal{F}_G \models \bigwedge_{i\in N}\left(\langle i,\sigma\rangle[i]\varphi \to [\sigma]\varphi\right)
\end{array}$$

Note that the first item, together with Theorem 14 says that σ is a best strategy for i in v_0 iff when σ leads us to a leaf z, and any deviation by i from σ leads to a leaf z', then in z' player i is not better off. Also note that the sub-game perfect notions are the global variants of the notions that hold in the root.

Taut	any classical tautology	$D!_\sigma$	$[\sigma]\varphi \leftrightarrow \langle\sigma\rangle\varphi$
K	$[\beta](\varphi \to \psi) \to ([\beta]\varphi \to [\beta]\psi)$	$F1$	$[\sigma,i]\varphi \to [\sigma]\varphi$
T_i	$[i]\varphi \to \varphi$	$F2$	$[\sigma,i]([\sigma',i']\varphi \leftrightarrow \varphi)$
4_i	$[i]\varphi \to [i][i]\varphi$	$F3$	$[\beta][\beta']([i]\varphi \to \psi) \vee [\beta''][\beta'''](([i]\psi \to \varphi)$
MP	$\vdash \varphi \to \psi,\ \vdash \varphi\ \Rightarrow \vdash \psi$	*Nec*	$\vdash \varphi \Rightarrow \vdash [\beta]\varphi$

Table 1. Axioms for Extensive Game Logic. The variables $\beta, \beta' \ldots$ range over the modalities $[i], [\sigma]$ and $[i,\sigma]$

The logic that comes with the semantics described here is dubbed *Extensive Game*

Logic in [35] (see Table 1). The axioms $Taut$ and K and the rules MP and Nec denote that we have a normal modal logic; T_i and 4_i reflect that the preference relation $\succeq_i$ is reflexive and transitive. Axiom $D!_\sigma$ says that R_σ is functional: any strategy profile σ prescribes a unique outcome. $F1$ expresses that if any deviation by i from σ leads to a result φ, then this is in particular the case if i sticks to σ_i, i.e., does not deviate. Property $F2$ reflects that any strategy $(\tau_i, \sigma_{\text{-}i})$ takes us to a leaf. And from any leaf, any further moves are void, i.e., $[\sigma', i']\varphi \leftrightarrow \varphi$ holds in it. Finally, $F3$ denotes a kind of *connectivity*. Correspondence theory (cf. Chapter 5) tells us that the modal scheme $\Box_1(\Box_3\varphi \to \psi) \vee \Box_2(\Box_3\psi \to \varphi)$ corresponds to the property that all R_1 and R_2-successors are R_3-connected: $\forall w, u, v : ((R_1wu \& R_2wv) \Rightarrow (R_3uv \text{ or } R_3vu))$. Hence $F3$ expresses that all $R_{\beta'} \circ R_\beta$ and $R_{\beta'''} \circ R_{\beta''}$-successors are $\succeq_i$-connected. By connectivity of $\succeq_i$, this axiom is sound; the fact that we are in the realm of game frames makes $F3$ also ensure this connectivity.

THEOREM 16. *([35, Theorem 4.1]) Extensive Game Logic as presented in Table 1 is strongly complete with respect to the semantics based on game models as defined on page 1092.*

3.3 Games as Process Models

Although [35] takes games in extensive form as its starting point, a little reflection on Figure 8 should convince the reader that the language only allows to reason about the *outcomes* of the game, not about *intermediate states*. Modal logic provides a wide range of languages, allowing to discriminate graphs, and hence games, on many levels of abstraction. This section discusses ideas from [85], by which also the following games G_1 and G_2 are inspired.

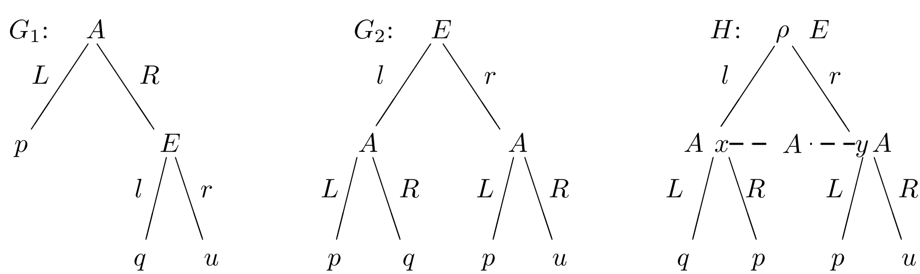


Figure 9. Two 'similar' games G_1 and G_2. H is an imperfect information game: see Section 6

Figure 9 represents two extensive game forms G_1 and G_2 for two players E and A, to which some propositional information has been added to the leaves. The question put forward in [85] is when two games are the same. When looking at the power of players, encoded in α-effectivity functions (cf. Section 2), the games G_1 and G_2 of Figure 9 *are* the same: the powers of E are $\{\{p, q\}, \{p, u\}\}$ while A is effective for $\{\{p\}, \{q, u\}\}$. Also, the two games represent *evaluation games* (cf. [37]) for two formulas that are equivalent, viz., $p \wedge (q \vee u)$ and $(p \wedge q) \vee (p \wedge u)$: verifier has a winning strategy for both formulas in exactly the same models. Hence, if we are interested in the outcomes of games only,

the games G_1 and G_2 can be coined the same. (We come back to the 'power level' for describing games in Section 10.)

On the other hand, there is admittedly a difference between the two games, if alone for the feeling that A can hand over control to E to achieve u in the left hand figure, whereas in the right hand side the converse looks more an appropriate description. Also, the two games differ by the mere fact that different players start in each. A property true in G_1 but not in G_2 for instance is that, if E is ever to move, he can guarantee u. Following [85], we see that when the turns and the moves of the players are at stake, the two extensive game models above are not equivalent.

When notions relating to the *process* of a game are important, formalisms, mainly arising from computer science, like process algebra, the Hoare-Dijkstra-Floyd calculus, dynamic modal logic and temporal logics naturally come to surface, each with its own semantic notions concerning equality of structures, like (finite) trace equivalence, observational equivalence or bisimulation. Although also the more recent *BDI-logics* (logics addressing Beliefs, Desires and Intentions, see also Chapter 18) for reasoning about multi-agent systems have similar concerns, at this moment there is not yet a dominant framework in which actions of agents (or moves of players), the powers of coalitions and the informational and motivational attitudes of the players are all treated on a par. We restrict ourselves to some semantical observations here.

Taking moves as atomic actions in a PDL-like logic (see also Chapter 12) a property true in the root of G_2 would be $\langle l \cup r\rangle[L \cup R](p \vee q)$: indicating that there is an execution of the choice $l \cup r$ so that every execution of the choice $L \cup R$ leads to either p or q. Apart from the choice operator $\cup$, one usually also has the constructs ; and * between programs, denoting sequential and iterative composition, respectively, and the test ? for propositions. To stay close to the previous section, where there are no labels for the player's choices, let us first mention that [35] assumes the following *atomic* actions (in our notation). For every player $i \in N$ we add an action $\mathbf{mo}_i$, where $R_{\mathbf{mo}_i}st$ is true iff if i is to move at s, he has a choice that leads to t. Next, for every strategy profile σ, relation $R_{\mathbf{step}(\sigma)}$ denotes the *one-step* transitions of σ, that is, $R_{\mathbf{step}(\sigma)}st$ iff $\sigma(s) = t$.

We now can distinguish the two models G_1 and G_2 with for instance the formula $[\mathbf{mo}_E][\mathbf{mo}_A \cup \mathbf{mo}_E]\bot$ which is only true in G_1, expressing that there are no moves possible after E has played. Following and specialising for instance [12], we can even identify formulas Φ_1 and Φ_2 such that Φ_i is true in a model M_i iff M_i is *bisimular* to G_i above. The idea is simple, constructing a formula $\varphi_{G,s}$ for every state s in a finite G inductively as follows. If s is a leaf, let $\varphi_{G,s}$ be the finite conjunction of literals over atoms true in s, together with the property that there are no moves to be made, i.e., $\varphi_{G,s} = \bigwedge_{G,s\models p} p \wedge \bigwedge_{G,s\not\models p} \neg p \wedge \bigwedge_{i\in N}[\mathbf{mo}_i]\bot$. Next, let t be a node such that for all its successors $s_1, \ldots s_k$ in the tree, φ_{G,s_h} $(h \leq k)$ is already defined, and suppose that i is to move in t. Then

$$\varphi_{G,t} =_{\text{def}} \bigwedge_{G,t\models p} p \wedge \bigwedge_{G,t\not\models p} \neg p \wedge \bigwedge_{j\neq i}[\mathbf{mo}_j]\bot \wedge \bigwedge_{h\leq k} \langle \mathbf{mo}_i\rangle\varphi_{G,s_h} \wedge [\mathbf{mo}_i](\bigvee_{h\leq k} \varphi_{G,s_h})$$

It will be clear from this construction, that for any extensive finite game G with root r_0 we have that for any model M and state s, $M, s \models \varphi_{G,r_0}$ iff M, s is bisimular to G, r_0. On game models, bisimulation boils down to *isomorphism*, but the notion is useful when reasoning about games as *graph automata* (see also Chapter 12 and [44]).

Observing that $\neg\langle\mathbf{step}(\sigma)\rangle\top$ characterises the leaves in the game tree, and using `while` φ `do` π as shorthand for $(\varphi? \ ; \ \pi)^*;\neg\varphi?$ ('as long as φ, perform π'), we can define, for every coalition $M = \{i_1, \ldots, i_m\} \subseteq N$ a program that, given σ, $N \setminus M$ adheres to it, but M is allowed to deviate:

$$\Pi(\sigma, M) =_{df} \texttt{while}\langle \mathit{step}(\sigma)\rangle\top \ \texttt{do} \ (\mathit{step}(\sigma) \cup \mathbf{mo}(i_1) \cup \cdots \cup \mathbf{mo}(i_m))$$

The operator $[\sigma]$ of the previous section can now be defined as $[\Pi(\sigma, \emptyset)]$, whereas $[\sigma, i]$ becomes $[\Pi(\sigma, \{i\})]$, enabling us to express the characterisations of (s.p.) Nash equilibrium and best response, as in the previous section.

But one does not have to reason about just one strategy. Let us define **end** as being in a leaf: $\mathbf{end} =_{\text{def}} \bigwedge_{i\in N}[\mathbf{mo}_i]\bot$. We can then express that, when every player i adheres to his strategy σ_i, the game will terminate in a state satisfying φ, using the formula $[(\bigcup_{i\in N}(\mathbf{mo}_i)?;\mathbf{step}(\sigma_\mathbf{i}))^*](\mathbf{end} \rightarrow \varphi)$. Every player i playing strategy σ_i of course induces a strategy profile σ, and as such, π can also be expressed in the framework of [35], but we can now also express properties of intermediate states. Let us define $\mathbf{plays}(i, \sigma_i)$ as $(\mathbf{mo}_i)?;\mathbf{step}(\sigma_\mathbf{i})$, and, for a set of strategies Σ_M, one σ_i for each $i \in M \subseteq N$, $\mathbf{Plays}(M, \Sigma_M)$ as $\bigcup_{i\in M}((\mathbf{mo}_i)?;\mathbf{step}(\sigma_\mathbf{i}))$, meaning that every agent i in M will play his strategy σ_i. Let us denote each agent j's atomic actions (or choices) with Ac_j. Also, let $\mathbf{any}(j)$ mean $(\mathbf{mo}_j)?;\bigcup_{a_j\in Ac_j}$, whereas $\mathbf{Any}(M) =_{\text{def}} \bigcup_{j\in M}\mathbf{any}(j)$. Then,

$$[(\mathbf{Plays}(M, \Sigma_M) \cup \mathbf{Any}(N \setminus M))^*]\varphi \tag{4}$$

expresses that coalition M can, by choosing the strategies Σ_M, ensure that, no matter what the other players $j \in N \setminus M$ will do, φ will invariantly be true. This implies that M has a strategy to ensure φ, which, as we will see in Section 11, is the basic expression of *ATL*. When abbreviating (4) as $\langle M, \Sigma_M\rangle\varphi$, that basic expression of *ATL* can be expressed as $\bigvee_{\Sigma_M}\langle M, \Sigma_M\rangle\varphi$, where Σ_M ranges over sets of strategies for players in M. It is interesting to note that every strategy in a finite game model is definable in *PDL*, using the characteristic formulae $\varphi_{G,s}$ for G, s: simply observe that every transition labelled with choice a_i from s to t can be denoted by $(\varphi_{G,s})?;a_i(\varphi_{G,t})?$. However, since in general there will be exponentially many strategies, this is arguably more of technical interest than of practical value.

3.4 Other Issues

Until now we have focused mainly on (Nash) equilibria and best response strategies. Recall from Zermelo's theorem (Theorem 7) that in every finite win-loss game for two players, exactly one player has a winning strategy. Let us denote player's i win by $\mathbf{win_i}$. Following [85] once more, we can define predicates $\mathbf{Win_i}$, meaning that, at the current node, i has a winning strategy (let $i \neq j$):

$$\mathbf{Win}_i \equiv (\mathbf{end} \wedge \mathbf{win}_i) \vee (\mathbf{turn}_i \wedge \langle\mathbf{any}(i)\rangle\mathbf{Win_i}) \vee (\mathbf{turn}_j \wedge [\mathbf{any}(j)]\mathbf{Win_i})$$

This hints at an inductive definition for $\mathbf{Win_i}$ using a *least fixed-point* schema

$$\mathbf{Win_i} =_{def} \mu P \cdot (\mathbf{end} \wedge \mathbf{win}_i) \vee (\mathbf{turn}_i \wedge \langle\mathbf{any}(i)\rangle P) \vee (\mathbf{turn}_j \wedge [\mathbf{any}(j)]P)$$

The reader immediately recognises the above as a formula in the *μ-calculus*, which is the topic of Chapter 12 of this handbook. As another example, the expression $\mu P \cdot (\mathbf{end} \wedge \varphi) \vee (\mathbf{turn}_i \wedge \langle \mathbf{any}(i) \rangle P) \vee (\mathbf{turn}_j \wedge [\mathbf{any}(j)]P)$ says that i has a strategy for guaranteeing a set of outcomes in which φ is true. Note that this is already expressible in a *PDL*-like logic: just choose $N = M = \{i, j\}$ in equation (4). In Section 11 we will discuss the logic *ATL* which is specifically designed to reason about what agents, and indeed, *coalitions* can guarantee to hold.

Still, the relation between the $\mu-$calculus and games is an interesting one. The calculus provides a very natural way, using its fixed point definitions, to reflect the equilibrium character of game-theoretic notions. Specifically the connections between the μ-calculus and games of possibly infinite duration are appealing: going back to an idea of [16], we know that *any* formula of the μ-calculus expresses the existence of a strategy in a certain game.

As seen above, if the goal of a player is to reach some desirable position in finite time, the set of positions that guarantee the win can be computed as a *least fixed point.* However, when the aim is to stay forever within a set of some safe positions, the winning set can be presented as a *greatest fixed point.* (This is very reminiscent of the distinction between *liveness* and *safety properties* used in computer science, as first introduced in [43]. See also [40] for a survey and Chapter 12 of this handbook.) More sophisticated winning conditions arise naturally in games modelling potentially infinite behavior of reactive systems. In general, mutually dependent least and greatest fixed point operators are necessary. [55] (from which the current two paragraphs borrow heavily), suggests that this interplay between least and greatest fixed points may well be the secret of the success of the μ-calculus: "... in contrast to first-order or temporal logic, the μ-calculus did not emerge by a formalization of the natural language". The μ-calculus also can benefit from game theory, since the game semantics reduces μ-calculus model checking to solving (parity) games: for more on this, see Chapter 12 of this handbook. A first impression of the complexity of such problems is given in Section 11. Another example of using games to settle complexity issues is provided in [12], where a *two person corridor tiling game* is used to prove the EXPTIME-hardness of PDL.

4 EPISTEMIC LOGIC

We saw in Section 2 that it makes sense to be explicit about the amount of information that each player has, at a given state of the game. Epistemic logic studies the notion of knowledge, and since [36], a mainstream in formal approaches to knowledge and belief is grounded in a possible world semantics. In the 1990's, these approaches were further developed in areas like computer science, cf. [21] – originally motivated by the need to reason about communication protocols – and artificial intelligence, cf. [49] and [51] – to reason about epistemic preconditions of actions. From the early days of game theory (cf. [4]) it has been recognised that the amount of knowledge that agents have is crucial in many solution concepts. But the formalisation of knowledge in game theory only took off since the late nineties, partially due to the TARK and LOFT events ([77, 45]).

The monograph [11] distinguishes between the notions of perfect/imperfect information on the one hand, and those of complete/incomplete information on the other. A game is of *perfect information* if the rules specify that the players always know 'where they are': for games in extensive form this means that each player is free in every node

	axioms and rules for $\mathbf{S}5_m$, where $i \leq m$		multi-agent notions
$A1$	any axiomatisation for propositional logic	$A6$	$E\varphi \leftrightarrow (K_1\varphi \wedge \cdots \wedge K_m\varphi)$
$A2$	$(K_i\varphi \wedge K_i(\varphi \to \psi)) \to K_i\psi$	$A7$	$(C\varphi \wedge C(\varphi \to \psi)) \to C\psi$
$A3$	$K_i\varphi \to \varphi$	$A8$	$C\varphi \to \varphi$
$A4$	$K_i\varphi \to K_iK_i\varphi$	$A9$	$C\varphi \to EC\varphi$
$A5$	$\neg K_i\varphi \to K_i\neg K_i\varphi$	$A10$	$C(\varphi \to E\varphi) \to (\varphi \to C\varphi)$
$R1$	$\vdash \varphi, \vdash \varphi \to \psi \;\Rightarrow \vdash \psi$		
$R2$	$\vdash \varphi \;\Rightarrow \vdash\; K_i\varphi$	$R3$	$\vdash \varphi \;\Rightarrow \vdash\; C\varphi$

Table 2.

to make a decision independent of that in other nodes. A game is of *complete information* if everything is known about the circumstances under which the game is played, like the probability that nature chooses a certain outcome, and who the opponent is and how risk-averse he is. In a game with incomplete information, players do not necessarily know which game they are playing, or who the other players are. Such games, although realistic, are up to now mainly the domain of a research area called *evolutionary game theory*, utilising theories of learning and evolutionary computation (see for an overview, [79]). Epistemic logic is nowadays widely used to express various degrees of imperfect information in a game; assumptions about the completeness of information are still often made on a meta-level.

Modal epistemic logic, the logic of knowledge, provides a very natural interpretation to the accessibility relation in Kripke models. For an agent i, two worlds w and v are connected (written R_iwv), if the agent cannot (epistemically) distinguish them. In other words, we have R_iwv if, according to i's information at w, the world might as well be in state v, or that v is compatible with i's information at w. Using this interpretation of access, R_i is obviously an equivalence relation. Readers familiar with game theory may be best acquainted with epistemic notions in this field as summarised in [5]. In that terminology, our set of states S is a space Ω of states of the world, our equivalence relation R_i is [5]'s partition $\mathcal{F}_i$ of Ω, and our *formulas* correspond to [5]'s *events*. Also, our equivalence relations connect in an obvious way to the partitions mentioned in Definition 8.

The epistemic modal language for m agents is obtained by allowing a modal operator K_i for every agent $i \leq m$, with $K_i\varphi$ meaning: agent i knows φ. The corresponding Kripke models are $M = \langle W, R_1, R_2, \ldots R_m, \pi\rangle$, with each R_i being an equivalence relation. Thus, we are in the realm of the multi-modal logic $\mathbf{S}5_m$, of which the axioms are summarised in Table 2. They express that knowledge is *closed under consequences* ($A2$), it is *veridical* ($A3$) and agents are both *positively* and *negatively* introspective ($A4$ and $A5$, respectively). Moreover, all agents know the $\mathbf{S}5_m$-theorems. Clearly, these properties represent *logically omniscient* agents (see also Chapter 18), and hence adopting this particular logic for the knowledge of players assumes an ideal case of perfect reasoners. For an overview of weakening the axioms to tackle logical omniscience, we refer the reader to [21, 49].

THEOREM 17. *We have the following facts for* $\mathbf{S5_m}$*, which is the logic summarised in*

the left hand side of Table 2.

1. *The system* $\mathbf{S5_m}$ *is sound and complete with respect to Kripke models in which every* R_i *is an equivalence relation. The complexity of the satisfiability problem for* $\mathbf{S5_m}$ *if* $m > 1$ *is* PSPACE-*complete (cf. [21]).*

2. *Taking* $m = 1$*, the logic* $\mathbf{S5_1}$ *is also sound and complete for the semantics where* R *is the* universal relation. *The complexity of the satisfiability problem for* $\mathbf{S5_1}$ *is the same as for propositional logic: it is NP-complete. Moreover, every formula* φ *is equivalent in* $\mathbf{S5_1}$ *to a formula without any nesting of modal operators (cf. [49]).*

EXAMPLE 18. Let us, as an example, consider the $S5_3$ model hexa, taken from [98], which focuses on the knowledge and ignorance of players in games with imperfect information, by introducing the notion of *knowledge games*, games in which the knowledge of the players, and the effect of their moves upon this knowledge, is described. In hexa, we have three players (1, 2 and 3) and three cards, each with a neutral side and a coloured face: r (red), w (white) or b (blue). If a player holds a card, he is the (initially only) player that knows its colour. See Figure 10 for the Kripke model hexa representing the knowledge of the players after the three cards have been dealt. The state rwb represents the deal where player 1 holds r, 2 holds w, and 3 holds b; this distribution is denoted in the object language as $\delta_{rwb} = r_1 \wedge w_2 \wedge b_3$. From now on, we will underline the 'actual' state of the model: hexa thus represents the knowledge of the players given that the actual deal is rwb. Reflexive access is not represented, thus it is understood that both rwb and rbw are 1-accessible from rwb: given that 1 has the red card, he does not know whether the deal is rwb or rbw.

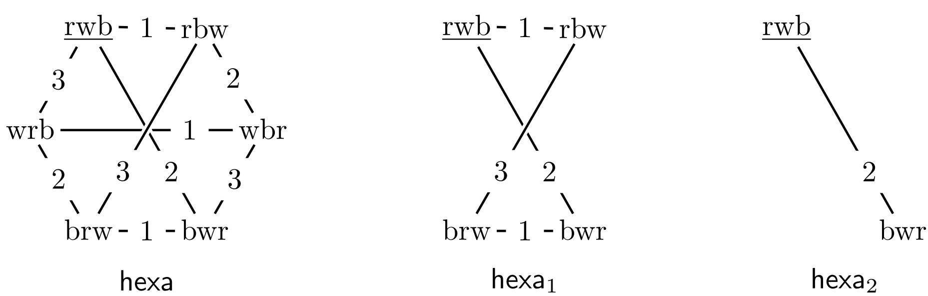


Figure 10. The initial model hexa; hexa$_1$ and hexa$_2$ are obtained using updates (cf. Section 7)

Interestingly, hexa not only tells us that every agent knows its own card (we have for instance hexa, $rwb \models K_1 r_1$, and, more generally, hexa, $c_1c_2c_3 \models K_i c_i$ ($c_i \in \{r, w, b\}, i \leq 3$), but also that everybody knows this. So, for instance, we have, and this is independent of the actual deal, hexa $\models K_1 \bigwedge_{c_2,c_3 \in \{r,w,b\}}((c_2 \rightarrow K_2 c_2) \wedge (c_3 \rightarrow K_3 c_3))$. And again, it is also the case that player 3 knows this. The aim of the game initiated by hexa is to find out the distribution of the cards, and we will return to the model when studying the dynamics of epistemics, in Section 7. $\dashv$

Indeed, the description of the situation that we gave is *common knowledge*, which is an intriguing multi-agent epistemic notion. Let us define 'everybody knows' ($E\varphi$) as is done in axiom $A6$ in Table 2 then the remaining axioms and rule on the right side of this table capture the intuition that common knowledge of φ models $E\varphi \wedge EE\varphi \wedge EEE\varphi \wedge .....$ Indeed, if we denote the axiom system represented in Table 2 by $\mathbf{S}5C_m$, then one easily checks that $\vdash_{\mathbf{S}5C_m} C\varphi \rightarrow E^n\varphi$, for arbitrary $n \in \mathbb{N}$, and, conversely, that $\{E^n\varphi \mid n \in \mathbb{N}\} \models C\varphi$: Semantically, the accessibility relation for R_E, with respect to which E-knowledge is the necessity, is $\bigcup_{i \leq m} R_i$, and then R_C (the relation for common knowledge) is the transitive closure R_E^* of R_E. We will denote any dual operator $\neg X \neg$ of X with $\hat{X}$.

EXAMPLE 19 (The muddy children). In this example the principal players are a father and k children, of whom m (with $m \leq k$) have mud on their foreheads. The father calls all the children together. None of them knows whether it is muddy or not, but they can all accurately perceive the other children and judge whether they are muddy. This all is common knowledge. Now the father has a very simple announcement (5) to make:

At least one of you is muddy. If you know that you are muddy, step forward. (5)

After this, nothing happens (except in case $m = 1$). When the father notices this, he literally repeats the announcement (5). Once again, nothing happens (except in case $m = 2$). The announcement and subsequent silence are repeated until the father's m-th announcement. Suddenly all m muddy children step forward!

Let us analyse the muddy children problem semantically, where we have 3 children. In Figure 11, the initial situation is modelled in twomud (we come back to a formal analysis of the story in Section 7). Worlds are denoted as triples xyz. The world 110 for instance denotes that child a and b are muddy, and c is not. Given the fact that every child sees the others but not itself, we can understand that agent a 'owns the horizontal lines' in the figure, since a can never distinguish between two states $0yz$ and $1yz$. Similar arguments apply to agents b and c.

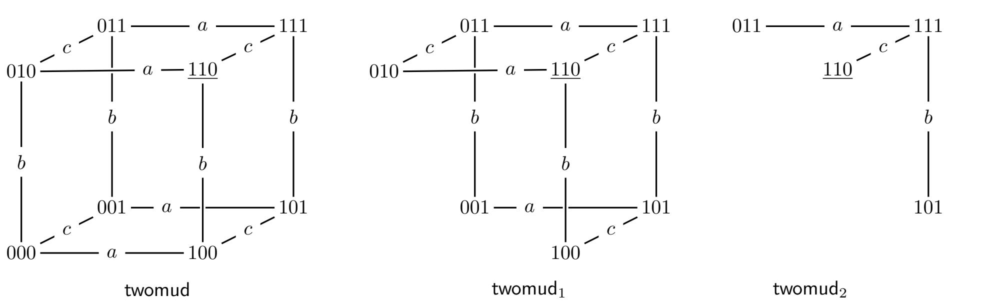


Figure 11. Muddy Children: initial situation twomud and after two updates (twomud$_1$ and twomud$_2$)

Let us see what epistemic truths we have in the state (twomud, s), with $s = 110$. The only propositional atoms we use are m_i ($i = a, b, c$) with meaning 'child i is muddy'. In state s, we then have for instance $\neg(K_a m_a \vee K_a \neg m_a)$ (agent a does not know whether it is muddy), and also $K_a(m_b \wedge \neg K_b m_b) \wedge K_a(\neg m_c \wedge \neg K_c \neg m_c)$ (a knows that b is muddy

without knowing it, and also that c is mudless without knowing that). Regarding group notions, we observe the following, in s. Let ℓ denote that at least one child is muddy ($\ell \equiv m_a \vee m_b \vee m_c$).

1. $E\ell \wedge \neg Em_a \wedge \neg Em_b \wedge \neg Em_c$; everybody knows that there is at least one muddy child, but nobody is known by everybody to be muddy

2. $K_c E\ell \wedge \neg K_b E\ell$; c knows that everybody knows that there is at least one muddy child, but b does not know that everybody knows at least one child to be muddy. To see the second conjunct, note that $\mathsf{twomud}, 100 \models \neg E\ell$, hence $\mathsf{twomud}, s \models \neg K_b E\ell$.

3. $\neg C\ell$; It is not common knowledge that there is at least one muddy child! This follows immediately from the previous item, but also directly from the model: one can find a path from $s = 110$ via 010 to 000, the latter state being one at which no child is muddy. One easily verifies that we have even $\mathsf{twomud} \models \neg C\ell$. $\dashv$

THEOREM 20 ([21]). *Let the logic* $\mathbf{S5C_m}$ *be summarised in Table 2. Then,* $\mathbf{S5C_m}$ *is sound and complete with respect to Kripke models in which every* R_i *is an equivalence,* R_E *(the relation for 'everybody knows') is* $\bigcup_{i \leq m} R_i$ *and* R_C *(the relation for common knowledge) is* R_E^**, i.e., the reflexive transitive closure of* R_E*. The complexity of the satisfiability problem for* $\mathbf{S5C_m}$ *if* $m = 1$ *is* PSPACE*-complete, and the complexity for* $\mathbf{S5C_m}$ *with* $m > 1$ *is* EXPTIME*-complete.*

As observed in [12, Chapter 6.8], the presence of a pair of modalities, one for a relation and the other for its reflexive transitive closure (in our case: E and C, respectively) is often –not always– an indication for entering the realm of EXPTIME-complexity results, since it enables one to force exponentially deep models, and to code the corridor tiling problem.

5 INTERPRETED SYSTEMS

Whereas the properties of knowledge as summarised in Table 2, especially that of negative introspection, have been under continuous debate and critique, the *Interpreted Systems* approach to knowledge as advocated by [21] in fact gives a *computationally grounded semantics* to the $\mathbf{S}5_m$ properties of knowledge. Rather than assuming the equivalence relation to be somehow given, in an interpreted system they naturally arise from the way a system is modeled. The idea is simple: in an interpreted system $\mathcal{I}$ we have m agents, or processors, each with its own local state s_i. A processor is aware of its own local state: two global states $\mathbf{s}$ and $\mathbf{s}'$ are the same for i if its local states in both coincide. And this notion of 'sameness' is an equivalence relation, yielding the $\mathbf{S}5_m$ properties in a natural way.

To formally define an interpreted system for m agents $\mathcal{I}_m$ we first give the notion of a global state. Let us assume that every agent i can be in a number of states L_i. Apart from the agents' local states, there is also a set of environment states L_e, which keeps track of, e.g., whether a communication line is up or down, or what the actual deal of cards is. A global state $\mathbf{s}$ is then a tuple $\langle s_e, s_1, s_2, \ldots s_m \rangle \in L_e \times L_1 \times L_2 \times \cdots \times L_m$. The set of global states of interest will be denoted $\mathcal{G} \subseteq L_e \times L_1 \times L_2 \times \cdots \times L_m$. Local states, both that of the agents and that of the environment, may change over time. A priori, there are no constraints on how the system may evolve: a *run* over $\mathcal{G}$ is a sequence

of states, or, rather, a function r from time $\mathbb{N}$ to global states. The pair (r, n) of a run and a time point is also referred to as a point. Let $r(n) = \langle s_e, s_1, s_2, \ldots s_m \rangle$ be the global state at time n in run r, then with $r_j(n)$ we mean s_j, where j ranges over $e, 1, \ldots, m$. Now, a *system* $\mathcal{R}$ over $\mathcal{G}$ is a set of runs over $\mathcal{G}$.

In general, formulas are now going to be interpreted in a point (r, n) in an interpreted system $\mathcal{I}$. To do so, we need to take care of atomic formulas and the epistemic operators. An *Interpreted System* $\mathcal{I} = (\mathcal{R}, \pi)$ over $\mathcal{G}$ is a system $\mathcal{R}$ over $\mathcal{G}$ with an interpretation π which decides for each point (r, n) and atom $p \in At$, whether p is true in (r, n) or not. Moreover, two points (r, n) and (r', n') are indistinguishable for i, written $(r, n) \sim_i (r', n')$, if $r_i(n) = r'_i(n')$, or, in other words, if i's local states in both points are the same. Definition (6) expresses that agent i knows φ in a point in $\mathcal{I}$ if φ is true *given agent* i's local information:

$$(\mathcal{I}, r, n) \models K_i\varphi \text{ iff } (\mathcal{I}, r', n') \models \varphi \text{ for all } (r', n') \sim_i (r, n) \tag{6}$$

We assume here that the interpretation function π only depends on the global state $r(n)$, and not the history of the point (r, n). This is in line with [21], but deviates from [30]. In the context of knowledge and linear time, the valid formulas are the same with and without this assumption: see [31] for an explanation of this. In practice, π will depend on just local information of a global state, denoting, e.g., whether a variable of processor i has a certain value, or whether player j holds the ace of hearts in a card game.

Of course, one may want to study epistemic logic for multi-agent systems in a static model $M_{\mathcal{G}} = \langle \mathcal{G}, \sim_1, \ldots, \sim_m, \pi \rangle$ based on a set of global states $\mathcal{G}$, and interpret formulas in global states themselves, rather than in a point in a run. If $\mathcal{G}$ equals the full Cartesian product $L_e \times L_1 \times L_2 \times \cdots \times L_m$, such models are called *hypercubes* in [47]. If we ignore the environment and only require that every combination of individual local states occurs we have a *full system*. Note that hexa of Figure 10 is a full system, whereas hexa_1 is obviously neither full nor a hypercube.

Full systems are appropriate classes of models to specify initial configurations of multi-agent systems, in which no agent has any information about any other agent, or about the environment. They are obviously $\mathbf{S5}_m$-systems, but interestingly enough they satisfy an additional property. It is not hard to see that the operator E for *everybody knows* from Section 4 semantically corresponds to interpreting E as the necessity operator of the relation that is the *union* of all the individual relations R_i: everybody knowns φ if nobody thinks it possible that $\neg\varphi$ is true. Note that $\hat{E}\varphi$ means that some agent considers it possible that φ: for some state that is $R_E = \cup_{i=1\ldots m} R_i$-accessible, φ is true. Axiom 7 then, on top of the axioms of $\mathbf{S5}_m$, is needed to axiomatise hypercubes and full systems (cf. [47]):

$$\bigwedge_{i=1\ldots m} \hat{E}\varphi_i \rightarrow \hat{E}\hat{E} \bigwedge_{i=1\ldots m} \varphi_i \tag{7}$$

In this scheme, φ_i is an i-local formula, to which we will come back shortly. Roughly, an i-local formula φ_i characterises i's knowledge: its truth value is constant within i's reachable states. The scheme then says that if we can reach, using R_E, a knowledge state for every agent, we can reach the 'combined knowledge state' in two steps.

When having more involved group notions of knowledge, full systems and hypercubes share some other specific properties. For instance, consider *distributed knowledge* $D\varphi$, in

which D is the necessity operator for the *intersection* of the individual relations R_i. This notion of knowledge arises when all agents would communicate with each other: if one agent rules out that s is a possible state, after communication nobody would consider s anymore. Obviously, in $\mathbf{S}5_m$, nobody rules out the current state, so that distributed knowledge in $\mathbf{S}5_m$ satisfies $D\varphi \to \varphi$. But in a full system with an empty environment, we also have the converse: $\varphi \to D\varphi$. This is easy to see: the only global state that is i-similar to $\mathbf{s} = \langle \emptyset, s_1, s_2, \ldots s_m \rangle$ is $\mathbf{s}$ itself!

Regarding common knowledge, in hypercubes and full systems it is either absent or else globally present: common knowledge is the same in all local states! This is easy also: if $C\varphi$ would hold in $\mathbf{s} = \langle s_e, s_1, s_2, \ldots s_m \rangle$ but not in $\mathbf{s}' = \langle s'_e, s'_1, s'_2, \ldots s'_m \rangle$, since the system is full, we find a state $\mathbf{t} = \langle \cdot, s_1, s'_2, \ldots, \cdot \rangle$. Recalling that R_C is the transitive closure of the R_i's, and that $R_1\mathbf{st}$ and $R_2\mathbf{ts}'$, we see that $\langle s_e, s_1, s_2, \ldots s_m \rangle$ and $\langle s'_e, s'_1, s'_2, \ldots s'_m \rangle$ must agree on $C\varphi$.

The logic of *local propositions* introduced in [20] connects the notion of accessibility in a system $M_{\mathcal{G}}$, in fact, in any epistemic model $M = \langle W, R_1, \ldots, R_m, \pi \rangle$, with a syntactic one. Let a proposition U in M just be a subset of W. It is an i-local proposition if for all $u, w \in W$ with $R_i uw$, we have $u \in U$ iff $w \in U$. In words: an i-local proposition is determined by i's local state, as is his knowledge. (In hexa for instance, r_1, denoting that 1 holds a red card, is 1-local.) For an atom p, say that $M \sim_p M'$ if the only difference between M and M' can be $\pi(p)$ and $\pi'(p)$. Define $M, w \models \exists_i p(\varphi)$ iff for some M' with $M' \sim_p M$, we have $M, w \models \varphi$, where p is an i-local propositional atom, i.e., $\pi(p)$ is an i-local proposition in M. Let $\Box$ be the universal operator, denoting what is true globally. Let q not occur in φ. Then:

$$M, w \models K_i\varphi \text{ iff } M, w \models \exists_i q(q \wedge \Box(q \to \varphi)) \tag{8}$$

(A similar 'reduction' can be given for distributed and common knowledge, see [20].) We have seen above in (7) that local propositions play a role in axiomatising hypercubes and full systems. But they have greater use: note that (8) implies that if the object language is rich enough to describe the local state of the agents, we can replace epistemic operators by occurrences of $\Box$ and local propositions. This idea is applied in [38] to 'reduce' model-checking of epistemic temporal properties to properties that can be handled by a 'conventional' model-checker SPIN that does not address knowledge explicitly: for a knowledge property $K_i\varphi$ to be checked in s, the user provides a local proposition q_i and the problem is then reduced to (1) checking whether q_i is indeed i-local, (2) whether it is true in s, and (3) whether the implication $q_i \to \varphi$ globally holds. To illustrate this, in hexa for instance, rather than saying, that $\mathsf{hexa}, rwb \models K_1(w_2 \vee w_3)$ (1 knows that the white card is owned by either player 2 or 3) we can stipulate: $\mathsf{hexa}, rwb \models r_1 \wedge \Box(r_1 \to (w_2 \vee w_3))$ (currently, player 1's local state reads 'red card', and globally, in such a situation either 2 or 3 holds the white card).

Let us now return to interpreted systems, where the dynamics is modelled through the notion of runs. Runs may look rather abstract, but following [21] one can think about them as being brought about by the agents while following a *protocol*, in which agents take certain *actions*. This is reminiscent of our notion of *strategies* in an extensive game: they restrict the space of all possible evolutions of the system. Having a language with operators for individual, distributed and common knowledge is still too poor to reason about interpreted systems: one easily shows that any two points (r, n) and (r', n') with the same global states $r(n) = r'(n')$ verify the same epistemic properties. Indeed, it is

natural to add *temporal* operators $\bigcirc$ (next time), $\square$ (always), $\diamondsuit$ (eventually) and $\mathcal{U}$ (until), with truth conditions like (for the formal interpretation of the other temporal operators, and a more extensive treatment of temporal logic, we refer to Chapters 11 and 17 of this Handbook):

$$(\mathcal{I}, r, n) \models \bigcirc\varphi \text{ iff } (\mathcal{I}, r, n+1) \models \varphi \tag{9}$$

With these operators, one *can* in general distinguish different points with the same global state: given a specific deal of cards for instance, it is perfectly well possible that in one play of the game player 1 will win, when in another play he will not: we can have $r(n) = r'(n')$, and $\mathcal{I}, r, n \models \diamondsuit\mathsf{win}_1$ but $\mathcal{I}, r', n' \not\models \diamondsuit\mathsf{win}_1$. In the temporal language, one can express that a certain property φ will occur infinitely often ($\square\diamondsuit\varphi$) or almost always ($\diamondsuit\square\varphi$). But the full character of the language of course comes to the fore in temporal-epistemic properties. Examples include $not\_cross\,\mathcal{U}\,K_i safe$, and $K_i\square((\neg K_j p \wedge \bigcirc K_j p) \rightarrow \bigcirc\bigcirc K_h K_j p)$ expressing that i will not cross the street until he knows it is safe, and that i knows that as soon as j learns that p, this will immediately be communicated to h.

Axioms for Linear Temporal Logic are given in Table 3: the operators $\diamondsuit$ and $\square$ can be defined in terms of $\bigcirc$ and $\mathcal{U}$. Regarding the soundness of the inference rule $R\mathcal{U}$, assume that $\mathcal{I} \models \varphi \rightarrow (\neg\psi \wedge \bigcirc\varphi)$. Since every occurrence of φ guarantees its truth in the next time, it is easy to see that we have $\mathcal{I} \models \varphi \rightarrow \square\varphi$. And, since φ comes with $\neg\psi$ we also have $\mathcal{I} \models \varphi \rightarrow \square\neg\psi$. Now, for an arbitrary χ, note that $\chi\,\mathcal{U}\,\psi$ being true would imply that ψ becomes true some time. But given φ, we just saw that ψ is always false, so we cannot have $\chi\,\mathcal{U}\,\psi$ when we have φ, hence $\mathcal{I} \models \varphi \rightarrow \neg(\chi\,\mathcal{U}\,\psi)$.

$T1$	$(\bigcirc\varphi \wedge \bigcirc(\varphi \rightarrow \psi)) \rightarrow \bigcirc\psi$	$T2$	$\bigcirc\neg\varphi \leftrightarrow \neg\bigcirc\varphi$
$T3$	$\varphi\,\mathcal{U}\,\psi \leftrightarrow \psi \vee (\varphi \wedge \bigcirc(\varphi\,\mathcal{U}\,\psi))$		
Nec	$\vdash \varphi \;\Rightarrow \vdash \bigcirc\varphi$	$R\mathcal{U}$	$\vdash \varphi \rightarrow (\neg\psi \wedge \bigcirc\varphi) \;\Rightarrow \vdash \varphi \rightarrow \neg(\chi\,\mathcal{U}\,\psi)$

Table 3. Axioms for Linear Temporal Logic **LTL.**

Note that in definition (6) in general we do not require that $n = n'$, so agents are not assumed to know what time it is. More generally, our definition allows the environment to change without any agent noticing this: two global states $\langle s_e, s_1, s_2, \ldots s_m\rangle$ and $\langle s'_e, s_1, s_2, \ldots, s_m\rangle$ look the same for all agents, but still may have a different environment. In fact, the definition of an interpreted system is so general that some agents i may sense changes in the environment (e.g., when $L_e \subseteq L_i$), while other agents j may not (when their local state L_j has 'nothing to do' with L_e).

Although in the general case no agent knows the time, in games it is usually assumed that all players at least know how many moves have been played. Indeed, this is, more often than not, assumed to be common knowledge. Games, and many other forms of competition and cooperation in multi-agent systems often assume that such a system is *synchronous*. To capture this, since the agents' knowledge is determined by their local state, we must encode a clock, or the number of passed 'rounds', in such local states. A system $\mathcal{R}$ is *synchronous* if each agent can distinguish different time points: if $n \neq n'$ then $(r, n) \not\sim_i (r', n')$, for any run r and r'. In other words, the local state of any agent must be different at different time points.

If we think of a run r as a sequence of global states in which each agent i is aware of its own local state $r(n)$ at time n, the question arises how much an agent *memorises* when going from $r_i(n)$ to $r_i(n+1)$. In the ideal case, i would 'remember' all local states $r_i(k)$ with $k \leq n$ when the time is currently n. In principle, this is the situation in games like chess and many other board and card games: each player has *perfect recall* of what he experienced during the game (although in practice, of course, humans and even machines may not have 'enough memory' for this). In a similar way as the requirement that i knows how many rounds have passed made us add this information to his local state, if we want the agent to remember exactly what has happened, from his perspective up to time n, we have to encode his previous local states $r_i(k)$ $(k < n)$ in his current local state at time n. Let therefore agent i's *local-state sequence* at the point (r, n) be his stutter-free local past $sflp_i(r, n) = \langle r_i(0), \pm r_i(1), \dots, \pm r_i(n-1)\rangle$, where $\pm r_i(x)$ means that $r_i(x)$ appears in the sequence iff it is different from its immediate predecessor in the sequence. Then, we say that i has *perfect recall in the system* $\mathcal{R}$ if $(r, n) \sim_i (r', n')$ implies that $sflp_i(r, n) = sflp_i(r', n')$, that is, if the agent remembers his local-state sequence, he has no uncertainty about what happened. We abstract from stuttering, since if the agent does not notice a change in his local state, he does not know 'how much is happening'— except in the synchronous case, when $sflp_i(r, n)$ is the complete sequence $\langle r_i(0), r_i(1), \dots, r_i(n-1)\rangle$.

Does perfect recall for agent i mean that $K_i\varphi \rightarrow \Box K_i\varphi$ is valid? Not in general, and mainly not because φ might refer to the current time. Knowing that 'today is Wednesday' does and should not imply that you always know that 'today is Wednesday'. Likewise, ignorance need not persist over time, and hence neither should the knowledge about it: even with perfect recall, $K_i\neg K_i p \rightarrow \Box K_i\neg K_i p$ should not hold: it would be equivalent to $\neg K_i p \rightarrow \Box\neg K_i p$ (note that in $\mathbf{S5}_m$, $\neg K_i\varphi$ is equivalent to $K_i\neg K_i\varphi$). Indeed, $\mathcal{I}, r, n \models K_i\neg K_i\varphi$ 'only' means that i knows to be ignorant in r at time n, this need not persist over time. Perfect recall would only require that at every point $(r, n+k)$, the agent knows that he *did not know* φ 'when the time was n'. A modal logic for belief revision in which one distinguishes operators B_0 for 'initial' belief (holding before the revision) and B_1 for 'new' beliefs (kept after the revision) is explored in [15]. The modal formulation of perfect recall of ignorance then becomes $\neg B_0\varphi \rightarrow B_1\neg B_0\varphi$. When we cannot distinguish between what *was* known and what *is* known, one might want to characterise the *stable* formulas for which $K_i\varphi \rightarrow \Box K_i\varphi$ holds in systems with perfect recall (see [21] and also discussions of 'only knowing' [91] and of (Un-)Successful Updates in Section 7). We will return to the property of perfect recall in Section 6: to capture perfect recall in synchronous systems we need the following property:

$$\textbf{PR} \qquad K_i\bigcirc\varphi \rightarrow \bigcirc K_i\varphi \quad (i = 1 \dots m)$$

We now give some technical results concerning the classes of systems discussed, in Theorem 21. To do so, let $\mathbf{S5}_m \oplus \mathbf{LTL}$ be the axioms for individual knowledge from Table 2, together with those for linear time from Table 3. This system assumes no interaction between knowledge and time. Also, let $\mathbf{S5}_m^C \oplus \mathbf{LTL}$ add the axioms for common knowledge (r.h.s. of Table 2) to this. Let $\mathcal{C}_m$ be the class of all interpreted systems for m agents, $\mathcal{C}_m^{sync}$ those that are synchronous, $\mathcal{C}_m^{pr}$ those that satisfy perfect recall, and let $\mathcal{C}_m^{sync,pr}$ be those synchronous interpreted systems that satisfy perfect recall.

THEOREM 21. *All the following results are from [21, Chapter 8], except for the first part of item 3, which is from [93]. Unless stated otherwise, assume $m \geq 2$.*

1. *$\mathcal{C}_m$ has $\mathbf{S5}_m \oplus \mathbf{LTL}$ as a sound and complete axiomatisation. The complexity of the validity problem for this class is PSPACE-complete. Adding common knowledge, $\mathbf{S5}_m^C \oplus \mathbf{LTL}$ completely axiomatises $\mathcal{C}_m$, but the validity moves to being EXPTIME-complete.*

2. *$\mathcal{C}_m^{sync}$: synchrony does not add anything in terms of axiomatisation or the validity problem: they are exactly as for $\mathcal{C}_m$.*

3. *$\mathcal{C}_m^{pr}$ is completely axiomatised by $\mathbf{S5}_m \oplus \mathbf{LTL} + \{\mathbf{KT}\}$, where $\mathbf{KT}$ is defined as $(K_i\varphi_1 \wedge \bigcirc(K_i\varphi_2 \wedge \neg K_i\varphi_3)) \rightarrow \hat{K}_i[K_i\varphi_1 \,\mathcal{U}\,(K_i\varphi_2 \,\mathcal{U}\, \neg\varphi_3)]$ (for a discussion, we refer to [93]). For $m = 1$, validity in $\mathcal{C}_m^{pr}$ is doubly-exponential time complete, otherwise non-elementary time complete. Adding common knowledge to the language makes the validity problem Π_1^1-complete. Hence, when common knowledge is present there is no finite axiomatisation for $\mathcal{C}_m^{pr}$; indeed, in this case there is not even a recursively enumerable set of axioms that is complete for validity in $\mathcal{C}_m^{pr}$.*

4. *$\mathcal{C}_m^{sync,pr}$ is completely axiomatised by $\mathbf{S5}_m \oplus \mathbf{LTL} + \{\mathbf{PR}\}$. The complexity of validity is non-elementary time complete. Adding common knowledge, we again get a complexity of Π_1^1 for the validity problem, and a negative result concerning finite axiomatisability for $\mathcal{C}_m^{sync,pr}$*

Our discussion of interpreted systems can only be limited. Rather than linear time, one may consider branching time logic, and apart from synchrony and perfect recall, one may consider properties with or without assuming a unique *initial state*, and with or without the principle of *no learning*– the 'converse of perfect recall'. Only these parameters all together yield 96 different logics: for a comprehensive overview of the linear case we refer to [31], and for the branching time case, to [97]. Moreover, where this section's exposition is mainly organised along the ideas in [21], there have been several other but related approaches to knowledge and time, or even knowledge and computation, of which we here only mention the distrtibuted processes approach of [60]. The recent paper [89] provides a general picture of different logics for knowledge and time, by giving a survey of decidability and undecidability results for several logics.

Although in general, the *model checking* problem is computationally easier than that of validity checking, for logics of knowledge and time, in particular those with perfect recall, the complexity of both tasks is often the same (see [94]). Work still progresses, both in the theoretical and the practical realm. We already mentioned an approach that 'reduces' epistemic temporal properties to temporal ones in order to use a 'standard' model checker. But model checkers that explicitly deal with an epistemic language are now rapidly emerging. Systems to model-check knowledge and time include the system MCK ([22]), DEMO ([104]) and the system MCMAS ([46]).

Recently, there has been a broad interest in model checking dynamics of knowledge in specific scenarios, like in [95] ('the dining philosophers'), and in [100] ('the sum and product problem'). Apart from model checking *epistemic properties*, it is also interesting to address the *realisability problem* (does there exist a *protocol* such that a given property is satisfied) and the *synthesis problem* (generate a protocol that satisfies a given constraint, if it exists). Space prohibits us to go into the details, we refer to [96].

6 GAMES WITH IMPERFECT INFORMATION

Now that we have seen how Kripke models are perfectly fit to represent games (Section 3) and imperfect information (Sections 4 and 5), let us spend some words on representing and reasoning about the combination of the two. In games with imperfect information, on which we will focus in this section, players *do* know what the rules of the games are, and who they play against, but they do not necessarily know 'where they are in the game'. In the game models of Section 3, this can be conveniently represented by using an indistinguishability relation for every player, as explained in Section 4. Game theorists call the members of each partition of such an $S5$-equivalence relation usually *information sets*. A game of perfect information would then just be the special case in which every information set contains exactly one node.

As a simple example of an imperfect information game, let us consider game H of Figure 9. We assume that we have the standard knowledge assumptions of $S5$, which semantically mean that the indistinguishability relation in that figure is an equivalence relation: however, we do not represent reflexive arrows, so that the only uncertainty in the game is represented by the dotted line labelled with player A. So, what is modelled in H is that player E makes a first move (l or r), and after that, A has to move, without knowing E's decision. In particular, we have $H, x \models K_A(\langle R\rangle p \vee \langle L\rangle p) \wedge \neg K_A\langle R\rangle p \wedge \neg K_A\langle L\rangle p$, in words: A knows he can guarantee p, but he does not know how! Recall that since a player is supposed to base his decision on the information at hand, we only consider *uniform strategies*, i.e., strategies σ which satisfy the following condition:

$$\forall s, t \in P^{-1}[\{i\}](R_i st \Rightarrow \sigma(s) = \sigma(t)) \tag{10}$$

Without this constraint on strategies, we would have that player A can enforce the outcome p in game H of Figure 9: $H, \rho \models [l, r]\langle L \cup R\rangle p$, which is counterintuitive: in order to achieve p, player A must play a different move in two situations that he cannot distinguish. Concluding: A has a strategy to ensure p in H, although we cannot expect him to play it, because the strategy is not uniform. But there is more to say about this. Suppose that in game H player E makes his move, and *then* we ask ourselves whether A has a uniform strategy to win. Surprisingly, he has! If E would play l, then A can use the uniform strategy σ_1, with $\sigma_1(x) = \sigma_1(y) = R$, and were E to play r, player A can fall back on the uniform strategy $\sigma_2(x) = \sigma_2(y) = L$. So, rather than just requiring uniform strategies for players to be used, we need an additional feature to distinguish winning situations from others. The notion that we are after seems to be closely related to the notions of *knowing-de dicto* and *knowing-de re*. The former expresses that player i knows that he has a winning strategy (if Ac_i represents i's set of actions, this would mean $K_i \bigvee_{a \in Ac_i}\langle a\rangle\mathbf{win_i}$), whereas the latter expresses he knows how to achieve it: $\bigvee_{a \in Ac_i} K_i\langle a\rangle\mathbf{win_i}$. For a further discussion on knowing-*de re* and knowing-*de dicto* in the context of extensive games (applying this to full strategies, rather than actions), see also [39].

The important difference between these two notions of knowledge, and its consequences for a theory of action, was already made in 1990 in [51] (and it goes back to the question what it means that 'A knows who B is', [36] and the general problem of 'quantifying in' into a knowledge formula, [69]). In the context of reasoning about knowledge and action [51] has been very influential, as it is still in the area of decision making in multi-agent systems. In order to cope with examples like 'in order to open a safe, you have to know

its key', [51] demonstrates the value of a possible world semantics, under the assumptions that terms are *rigid designators*. Although the language used in [51] is that of first order logic, one easily recognises properties like *perfect recall* and *no learning* (see Section 5 and 6.1) in [51]'s *noninformative action*.

[51] argues that the assumption of perfect information has been the prominent one in planning in Artificial Intelligence up to the nineties. What is missing, in the first place, is an analysis of *epistemic pre-conditions* before executing a plan: does the agent have the information necessary to carry out (the next step of) the plan? This question has been taken up by computer scientists by coming up with the notion of *knowledge programs*, (cf. [21, Chapter 7]) in which objective test conditions in a program are replaced by epistemic conditions. Apart from specifying epistemic pre-conditions, [51] argues to include *epistemic post-conditions*, in order to facilitate the agent to reason about how to acquire the information that is lacking to execute a sequential plan successfully. All these notions are crucial in the context of game playing, as well.

Where we thus far 'only' imposed the $S5$ principles on imperfect information games, [83] mentions several properties that one could require 'on top' of this, some even related to incomplete information games, like $\mathbf{turn_i} \rightarrow C\mathbf{turn_i}$ (it is common knowledge whose turn it is) and $\langle a \rangle \top \rightarrow C\langle a \rangle \top$ (it is common knowledge which moves can be played, in any node).

If information sets were only used to impose players to stick to uniform strategies (10) one only needs to specify knowledge-accessibility for player i at i's decision nodes. But, although this seems indeed to be common practice in game theory, the full machinery of $S5$ allows for much finer structure. And there are many more assumptions in game theory that seem dominant (but maybe easily relaxed), like the one that made us draw a *horizontal* line as indistinguishability in Figure 9, suggesting that players know how long the game has been played for. In general, we can require that agents don't know what has happened, or whose turn it is. Rather than systematically describe all the options, in the next section we focus on one specific property, and show how a modal analysis may be of help.

6.1 Case study: Perfect Recall

The principle of perfect recall in a game captures that players have some memory about what happened. Formally, in a dynamic logic setting, it is expressed as (see [83] for further details):

$$\begin{array}{llr} (i) & (\mathbf{turn_i} \wedge K_i[a]\varphi) \rightarrow [a]K_i\varphi & \quad and \\ (ii) & (\neg\mathbf{turn_i} \wedge K_i[\cup_{b\in B}]\varphi) \rightarrow [\cup_{b\in B}]K_i\varphi & \end{array} \tag{11}$$

where a is any action of player i, and B is the union of all actions of the other players. In words: if player i knows that doing a will lead to φ, then after having done a, he knows φ. Note that clause (i) of (11) is not realistic for *specific* moves of the opponent: in game H of Figure 9 for instance, we have $H, \rho \models K_A[l]\langle R\rangle p$, but not $H, \rho \models [l]K_A\langle R\rangle p$. Also, it assumes that players are aware when others make their move (cf. synchrony in Section 5).

Semantically, the subformula $K_i[a]\varphi \rightarrow [a]K_i\varphi$ of (11) corresponds to the following (which is even more apparent from the dual of (11), i.e. $\langle a\rangle\hat{K}_i\varphi \rightarrow \hat{K}_i\langle a\rangle\varphi$):

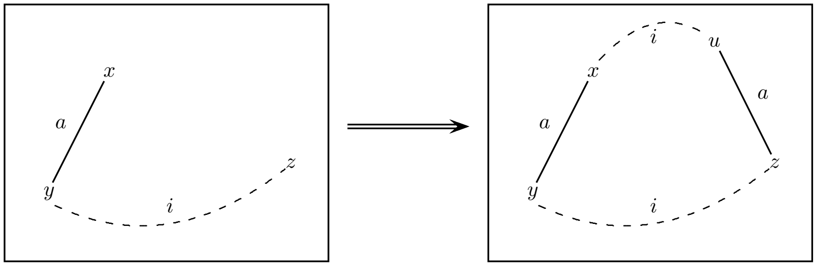


Figure 12. The commuting condition pictured. The arrow stands for implication

$$\forall xyz((R_a xy \ \&\ R_i yz) \rightarrow \exists u(R_i xu \ \&\ R_a uz)) \tag{12}$$

This 'commutation property' is also depicted in Figure 12, and guarantees that ignorance, or better indistinguishability, cannot be generated spontaneously: if a player i cannot distinguish, after doing action a, between state z and y, then both states must be the result of performing a in indistinguishable states.

For illustrative purposes, let us now look how [14] generalises these notions (over more than just actions), to characterise *von Neumann games*. Let $s \prec t$ in a game tree denote that there is a path (labelled with choices) from s to t. Next, let $\ell(x)$ denote the number of predecessors of node x according to $\prec$. Then, an extensive game is called von Neumann if

$$\forall xy(R_i xy \ \&\ x, y \in P^{-1}[\{i\}] \Rightarrow \ell(x) = \ell(y)) \tag{13}$$

This implies that in a von Neumann game, a player who has to move knows how many moves have already been played. A game satisfies *Memory of Past Knowledge* (MPK) if

$$\forall xyz((x \prec y \ \&\ R_i yz) \rightarrow \exists u(R_i xu \ \&\ u \prec z)) \tag{14}$$

Note that MPK is weaker than perfect recall in that it abstracts from the specific action taken. In [14] it is shown that perfect recall is equivalent to the conjunction of Memory of Past Knowledge and Memory of Past Actions (in which a player remembers which actions he has taken in the past, not necessarily in which order).

THEOREM 22. *Let G be an extensive form game. Then, the following are equivalent:*

1. *G satisfies MPK*
2. $\forall i, \forall x, y(R_i xy \Rightarrow \ell(x) = \ell(y))$

Note that condition 2 says that it is in fact common knowledge that G is a von Neumann game. The direction (1) $\Rightarrow$ 2 is proven in [14], the other direction in [9]. The modal characterisation of MPK is given in [14] by using temporal operators. The Past operator P is for instance interpreted as follows: $G, s \models P\varphi$ iff for all t for which $t \prec s$, we have $G, t \models \varphi$. Similarly, the operator $\Box$ refers to the future (for more on temporal modal logic, we refer the reader to Chapter 11 of this handbook). Then, MPK is characterised by

$$PK_i\varphi \rightarrow K_i PK_i\varphi \tag{15}$$

As a scheme, (15) is equivalent to $K_i\varphi \rightarrow \Box K_i PK_i\varphi$.

6.2 Outlook

The upshot of the exercise in the previous section is not so much the specific correspondence results, but rather to demonstrate the elegance and suitability of the modal machinery to reason about notions such as perfect recall. The latter notion also has received considerable attention in the computer science literature, where, in the context of synchronous systems, it is defined as $K_i \bigcirc \varphi \rightarrow \bigcirc K_i \varphi$, see Section 5. (Perfect recall is called 'no-forgetting' in the seminal paper [29]). The other direction of non-forgetting, i.e., $\bigcirc K_i \varphi \rightarrow K_i \bigcirc \varphi$ is called 'no learning', and comes with a similar commutation diagram as that in Figure 12.

Unfortunately, such commuting diagrams that enforce regularities on the underlying models have in general an adverse effect on the complexity of checking satisfiability for such logics. Having a grid-like structure in models for a logic enables one to encode Tiling Problems in them, which then can be used to demonstrate that in the worst case satisfiability becomes undecidable (see also Chapters 3 on complexity and 7 on decision problems). In fact, [29] shows that only assuming no-learning (in a context of at least two knowers and allowing common knowledge), the validity problem is highly undecidable. In words of [89]: "Trees are safe" and "Grids are Dangerous". We refer to that paper for a survey of several (un-)decidability results for temporal epistemic logics. For a restricted set of related results, we refer to Theorem 21 in this Chapter and its succeeding paragraphs.

7 DYNAMIC EPISTEMIC LOGIC

The framework for epistemic logic as presented in Section 4 elegantly allows for reasoning about knowledge (and, in particular, higher order knowledge: knowledge about (other's) knowledge), but as such it does not allow to deal with the *dynamics of epistemics*, in which one can express how certain knowledge changes due to the performance of certain actions, which by itself can be known or not. The notion of run in an interpreted system (Section 5) explicitly allows for such dynamics. In this section, we look at dynamic epistemic logic where the actions themselves are epistemic, like a revision due to a public announcement or a secret message. The famous paper [2] put the *change of information*, or belief revision, as a topic on the philosophical and logical agenda (cf. Chapter 18). This AGM tradition typically studies how a belief or knowledge set should be modified, given some new evidence. Well-studied examples of such modification are *expansion*, *contraction* and *revision*, which are of type $2^{\mathcal{L}} \times \mathcal{L} \rightarrow 2^{\mathcal{L}}$, i.e., they transform a belief set K given new evidence φ into a new belief set K', where the belief sets are subsets of the propositional language. The publication of [2] generated a large stream of publications in belief revision, investigating the notion of epistemic entrenchment, the revision of (finite) belief bases, the differences between belief revision and belief updates, and the problem of iterated belief change (for more on belief revision, refer to Chapter 18 in this handbook).

However, in all these approaches the dynamics are studied on a level *above the informational level*: the operators for modification are not part of the object language, and they are defined on (sets of) propositional formulas in $\mathcal{L}$. Hence, it is impossible to reason about change of agents' knowledge and ignorance within the framework, let alone about the change of other agents' information. This section describes approaches where the changing epistemic attitudes find their way into the object language.

The notion of a run in an interpreted system, together with the availability of temporal operators in the language (Section 5) facilitates reasoning about the dynamics of an agent's knowledge. A run is typically generated by a protocol, which usually represents a standard computer program. The pioneering work in [51] also studies the relation between actions and knowledge: there the emphasis is on epistemic preconditions that are needed to perform certain actions in the world, such as knowing a key-combination in order to open a safe. From the point of view of expressivity, one can say that the work on interpreted systems enables one to reason about the (change of) knowledge over time, and adding actions to the language, one can also reason about the change of knowledge brought about by performing certain plans. This enables one to express properties like *perfect recall* and *no learning* discussed in previous sections.

This section sketches approaches that not only "dynamise the epistemics", but also "epistemise the dynamics": the actions that (groups of) agents perform are epistemic actions. Different agents may have different information about which action is taking place, including higher-order information. Perfect recall would then rather look like $K_i[\alpha]\varphi \rightarrow [K_i\alpha]K_i\varphi$: "if player i knows that when j chooses 'right' this offers i a possible win, only after i *knows* that j *does* move 'right', i is aware of his profitable situation"! The rather recent tradition often referred to as *Dynamic Epistemic Logic*, treats all of knowledge, higher-order knowledge, and its dynamics on the same level. Following a contribution of 1997 [23], a stream of publications appeared around the year 2000 ([48, 98, 84]) and a general theory only now and partially emerges. In retrospect, it appeared that an original contribution of [68] from 1989 was an unnoticed ancestor of this stream of publications. This section is too short to discuss all those approaches, we will, for homogeneity, mainly follow [102, 101], and [7]. We start by considering a special case of updates.

7.1 Public Announcements

Public announcements are a simple and straightforward, but still interesting epistemic action: the idea behind a public announcement of φ is that all players are updated on φ, and they all know this, and they all know that they know this, etc. Given a group of players N, the language $\mathcal{L}_N^{pa}$ for public announcements adds a modality $[\chi]\psi$ on top of the epistemic language with operators $K_i(i \in N)$. If the common knowledge operator C is also allowed, we refer to the language as $\mathcal{L}_N^{pa}(C)$. The interpretation of $[\chi]\psi$ reads: "after truthful public announcement of χ, it holds that ψ". Note that both χ and ψ are typical members of $\mathcal{L}_N^{pa}$ or $\mathcal{L}_N^{pa}(C)$: announcements can be nested.

The semantics of $[\chi]\psi$ is rather straightforward: it is true in (M, s) if, given that χ is true in (M, s), ψ is true in s if we 'throw away' all the states in which χ is false. To achieve this, we define $M_{|\chi}$ as that submodel of M that consists of all points in which χ is true. More formally, given $M = \langle W, R_1, R_2, \ldots R_m, \pi\rangle$, the model $M_{|\chi} = \langle W', R'_1, R'_2, \ldots R'_m, \pi'\rangle$ has as its domain all the χ states: $W' = \{w \in W \mid (M, w) \models \chi\}$, and the primed relations and valuation in M' are the restrictions of the corresponding relations and valuation in M to W'. Then, we define

$$M, s \models [\chi]\psi \text{ iff } (M, s \models \chi \Rightarrow M_\chi, s \models \psi). \tag{16}$$

EXAMPLE 23. (Example 18 ctd.) In the miniature card game hexa (Example 18), suppose that in (hexa,rwb), player 1 publicly announces that he does not possess card w, i.e.,

$\varphi = \neg w_1$. Then, the resulting model is $\mathsf{hexa}_{|\varphi} = \mathsf{hexa}_1$ of Figure 10: all the deals in which 1 *does* have the white card are removed. Note that we have: $\mathsf{hexa}, rwb \models [\neg w_1]K_3 r_1$, and even $\mathsf{hexa}, rwb \models [\neg w_1]K_3(r_1 \wedge w_2 \wedge b_3)$, saying that after 1's announcement, player 3 knows the exact deal. Note that this is not true for player 2: $\mathsf{hexa}, rwb \models [\neg w_1]\neg K_2 r_1$, since, after the announcement $\neg w_1$, player 2 still considers it possible that 1 has b. Player 2 knew already the truth of the announcement. Still, he 'learns' from it:

$$\mathsf{hexa}, rwb \models K_2 \neg K_3(r_1 \wedge w_2 \wedge b_3) \wedge [\neg w_1]K_2(K_3(r_1 \wedge w_2 \wedge b_3) \vee K_3(b_1 \wedge w_2 \wedge r_3))$$

This expresses that initially, in rwb, 2 knows that 3 does not know the current deal (described by $r_1 \wedge w_2 \wedge b_3$), but after 1's announcement $\neg w_1$, 2 knows that 3 knows the deal. Note that 1 does not learn the same as 2: player 1 cannot be sure that 3 learns the deal from the announcement $\neg w_1$, since, according to 1, it might be the case that 3 holds w, in which case 3 would not learn the deal from the announcement.

Public announcements can be made iteratively: the model hexa_2 is obtained from hexa_1, by letting 3 make the public announcement "I know the deal!". More formally, let $\mathsf{knowsdeal}(i)$ be $\bigvee_{c,d,e \in \{r,w,b\}} K_i(c_1 \wedge d_2 \wedge e_3)$. Then, 1 learns the deal after 3 announces that he learned, but 2 does not (let δ be the actual deal $(r_1 \wedge w_2 \wedge b_3)$):

$$\mathsf{hexa}, rwb \models [\neg w_1][\mathsf{knowsdeal}(3)](K_1\delta \wedge \neg K_2\delta)$$

This can be formally verified by inspecting Figure 10, but is also intuitively correct: if 1, holding r, announces that he does not possess w, then he knows that this is either informative for 2 (in case 3 has w, i.e., in rbw) or for 3 (in rwb). Since 3 subsequently announces he learned the deal, 1 finds out the real situation is rwb. Similarly, 2 does not learn the deal from this "dialogue", he conceives it still possible that the real deal is bwr. However, as we saw above, 2 still learns something (i.e., about the knowledge of others: after the first announcement 2 learns that 3 knows the deal and after the second 2 learns that also 1 knows the deal). $\dashv$

As for an axiomatisation of public announcements, the logic $\mathbf{S5}_N^{pa}$ is obtained by adding the left-hand side of Table 4 to $\mathbf{S5}_m$. The logic $\mathbf{S5}_N^{pa}(C)$, which also incorporates common knowledge, is axiomatised by the union of Tables 2 and 4.

	axioms and rules for $\mathbf{S5}_N^{pa}$		additional rule for $\mathbf{S5}_N^{pa}(C)$
A11	$([\chi]\varphi \wedge [\chi](\varphi \rightarrow \psi)) \rightarrow [\chi]\psi$		
A12	$[\chi]p \leftrightarrow (\chi \rightarrow p)$	*R5*	From $\vdash \varphi \rightarrow [\chi]\psi$
A13	$[\chi]\neg\psi \leftrightarrow (\chi \rightarrow \neg[\chi]\psi)$		and $\vdash \varphi \wedge \chi \rightarrow E\varphi$
A14	$[\chi]K_i\psi \leftrightarrow (\chi \rightarrow K_i[\chi]\psi)$		infer $\vdash \varphi \rightarrow [\chi]C\psi$
R4	$\vdash \varphi \Rightarrow \vdash [\chi]\varphi$		

Table 4. Public announcements without (left) and with common knowledge

Axiom $A11$ and rule $R4$ characterise $[\chi]$ as a normal modal operator. The other axioms have the general form $[\chi]\varphi \leftrightarrow (\chi \rightarrow \varphi')$; the second appearance of χ indicates that only behaviour of *successful updates* is specified. Note that the general form is equivalent to $(\chi \wedge ([\chi]\varphi \leftrightarrow \varphi')) \vee (\neg\chi \wedge [\chi]\varphi)$, which, by the fact that $\vdash \neg\chi \rightarrow [\chi]\varphi$, is equivalent to $(\chi \wedge ([\chi]\varphi \leftrightarrow \varphi')) \vee \neg\chi$.

Keeping this in mind, axiom $A12$ (also called *atomic persistence*) assures that atomic facts (and, hence, objective properties, not involving any knowledge) are not affected by public announcement: they do not change the world. According to $A13$, public announcements are partial functions: under the condition that the announced formula is true, its announcement induces a unique outcome. Finally, axiom $A14$ relates individual knowledge to public announcements: from right to left it is a variant of the earlier mentioned *perfect recall*, the other direction is a conditionalised *no learning* property.

The straightforward generalisation of $A14$ to a logic with common knowledge would read $[\chi]C\psi \leftrightarrow (\chi \to C[\chi]\psi)$. However, such a principle is not valid. Consider the models M and M' of Figure 13 (taken from [101]). First of all, let 10 denote a world in which p is true, and q is false, and similarly for 11 and 01. Then, in model M of Figure 13, we have $M, 11 \models [p]Cq$, since in the updated model $M' = M_{|p}$, we have $M', 11 \models Cq$ (in M', world 11 is only $a-$ and $b-$ accessible to itself). At the same time, we have $M, 11 \not\models (p \to C[p]q)$, in particular $M, 11 \not\models C[p]q$. This is so since in 11 there is a world R_C-accessible (to wit, 10), in which a public announcement of p would lead us to the disconnected part 10 of M', where we have $M', 10 \not\models q$, so that $M, 10 \not\models [p]q$, which justifies the claim $M, 11 \not\models C[p]q$.

Figure 13. A counter model for $[p]Cq \leftrightarrow (p \to C[p]q)$

In order to obtain common knowledge through a public announcement, rule $R5$ from Table 4 must be used. The soundness of this rule is typically proven using induction on the R_C-path to ψ in an updated model $M_{|\chi}$.

THEOREM 24 ([68, 8]). *The logics* $\mathbf{S5}_N^{pa}$ *(without common knowledge) and* $\mathbf{S5}_N^{pa}(C)$ *(with common knowledge) as defined in Table 4 are sound and complete with respect to the semantics with key condition (16) on top of the epistemic semantics as given in Section 4.*

In fact, dynamic epistemic logic $\mathbf{S5}_N^{pa}$ can be reduced to its static counterpart $\mathbf{S5}_{|N|}$, by employing the following translation T:

$$
\begin{array}{lcl}
T([\chi]p) & = & T(\chi) \to p \\
T([\chi](\varphi \wedge \psi)) & = & T([\chi]\varphi) \wedge T([\chi]\psi) \\
T([\chi]\neg\psi) & = & T(\chi) \to \neg T([\chi]\psi) \\
T([\chi]K_i\psi) & = & T(\chi) \to K_i T([\chi]\psi)
\end{array}
$$

The equivalence between φ and $T(\varphi)$ follows immediately from the axioms of Table 4, and it is also easy to see that $T(\varphi)$ has no occurrences of the $[\chi]$ operator: they are all replaced by implications $\chi \to p$ for certain atoms p. This feature can be used to obtain completeness of $\mathbf{S5}_N^{pa}$ ([68]), but for the case of $\mathbf{S5}_N^{pa}(C)$, the completeness proof is much more involved ([8]).

Unsuccessful Updates

The intuition of a public announcement $[\chi]$ is that it produces common knowledge of the announced fact χ. Remarkably enough, it is not always the case that $[\chi]C\chi$. As an example, take the model M of Figure 13, where in 11 the atom p holds, but a is ignorant about it. The public announcement of this very fact (i.e., $p \wedge \neg K_a p$, which could be uttered by player b, since he knows it) however, leaves a with a difficult, if not impossible task to update his knowledge; it is hard to see how to simultaneously incorporate p and $\neg K_a p$ into his knowledge.

DEFINITION 25 ((Un-)successful Formulas and Updates). A formula χ is *successful* if $\models [\chi]\chi$. Otherwise, it is *unsuccessful.* Moreover, χ is a *successful update* in M, s if $M, s \models (\chi \wedge [\chi]\chi)$, it is an *unsuccessful update* if $M, s \models (\chi \wedge [\chi]\neg\chi)$. An update with χ is *publicly successful* in M, s if $M, s \models \chi \wedge [\chi]C\chi$. $\dashv$

Which formulas are unsuccessful, and which are successful? This question was raised in [82], and some first answers are given in [86] and [99]. Typically, only formulas involving ignorance can be unsuccessful. Hence, propositional formulas, involving no epistemic operators are always successful. Secondly common knowledge formulas are successful, by merits of the validity $\vdash [C\varphi]C\varphi$. The paper [99] identifies a fragment of the language, $\mathcal{L}_N^{u0}$ that is preserved under 'deleting states':

$$\varphi \in \mathcal{L}_N^{u0} ::= p \mid \neg p \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid K_i\varphi \mid C\varphi \mid [\neg\varphi]\psi$$

The fragment $\mathcal{L}_N^{u0}$ is preserved under submodels, from which it follows that for any $\varphi \in \mathcal{L}_N^{u0}$ and any ψ, $\vdash \varphi \rightarrow [\psi]\varphi$. As a consequence, the language $\mathcal{L}_N^{u0}$ is successful. After presenting these partial results, and before giving an example of (un-)successful updates, we mention the following fact about updates:

THEOREM 26 ([86]). *In every model, every public announcement is equivalent to a successful one.*

EXAMPLE 27. (Example 19 ctd., [99]) Consider model twomud from Figure 11. Let us abbreviate $(m_a \vee m_b \vee m_c)$ to muddy. Then model $(\mathsf{twomud}_1, 110)$ is the model that one obtains when publicly announcing muddy in $(\mathsf{twomud}, 110)$, i.e., after (5) is announced. One easily checks that muddy is a publicly successful update in this state: $\mathsf{twomud}, 110 \models [\mathsf{muddy}]C\mathsf{muddy}$ (note that, since muddy is a member of the submodel-preserving language $\mathcal{L}_N^{u0}$, it is even a successful *formula*).

Note that $\mathsf{twomud}, 100 \models [\mathsf{muddy}]K_a m_a$: if a is the only muddy child, he knows about his muddiness after the announcement (5) that there is at least one muddy child. Let $\mathsf{knowmuddy} = \bigvee_{i \in \{a,b,c\}}(K_i m_i \vee K_i \neg m_i)$ (at least one child knows about its muddy state). Now, although we have $\mathsf{twomud}_1, 110 \models \neg\mathsf{knowmuddy}$, we have $\mathsf{twomud}_1, 110 \models [\neg\mathsf{knowmuddy}]\mathsf{knowmuddy}$. In other words, when the father makes his announcement (5) for the second time, we interpret this as an announcement of $\neg\mathsf{knowmuddy}$ (since father makes his remark for the second time, this is a public announcement that no child stepped forward after the first utterance of (5), or, in other words, no child knows yet about its muddiness). Since $\mathsf{knowmuddy}$ is true in twomud_1 in the states 001, 010 and 100, these states are removed after the announcement $\neg\mathsf{knowmuddy}$, giving us model twomud_2 of Figure 11.

To further explain the story, one easily verifies $\mathsf{twomud}_2, 110 \models C\mathsf{atleasttwomuddy}$, with $\mathsf{atleasttwomuddy}$ having its obvious interpretation $\bigvee_{i,j \in \{a,b,c\}}^{i \neq j}(m_i \wedge m_j)$. But if this

is common knowledge in $\mathsf{twomud}_2, 110$, in particular children a and b know this: since they only see one other muddy child, they conclude that they are muddy themselves and hence step forward. We have seen that $\mathsf{twomud}, 110 \models \mathsf{muddy} \wedge [\mathsf{muddy}]\mathsf{Cmuddy}$, and also that $\mathsf{twomud}, 110 \models [\mathsf{muddy}]\ [\neg\mathsf{knowmuddy}]C\mathsf{knowmuddy}$. In other words, $\neg\mathsf{knowmuddy}$ is an unsuccessful update in $\mathsf{twomud}_1, 110$. Note that this is indeed a local notion: the same announcement $\neg\mathsf{knowmuddy}$ would have been successful in $(\mathsf{twomud}_1, 111)$. ⊣

7.2 General Updates

Public announcements play an important role in games: putting a card on the table, rolling a die, and moving a pawn on the chess board can all be considered as examples. However, in many situations much more subtle communication takes place than a public announcement. Consider the card game hexa in which player 1 shows player 2 his card. Obviously, this is informative for 2: he even learns the actual deal. But, although 3 does not see 1's card, he certainly obtains new information, viz. that 2 learns the deal. And 1 and 2 also get to know that 3 learns this!

The following – possibly simplest – example in the setting of multi-agent systems (two agents or players, one atom) attempts to demonstrate that the notions of higher-order information and epistemic actions are indeed non-trivial and may be subtle.

> *Anne and Bert are in a bar, sitting at a table. A messenger comes in and delivers a letter that is addressed to Anne. The letter contains either an invitation for a night out in Amsterdam, or an obligation to give a lecture instead. Anne and Bert commonly know that these are the only alternatives.*

This situation can be modelled as follows: There is one atom p, describing 'the letter invites Anne for a night out in Amsterdam', so that $\neg p$ stands for her lecture obligation. There are two agents 1 (Anne) and 2 (Bert). Whatever happens in each of the following action scenarios, is publicly known (to Anne and Bert). Also, assume that in fact p is true.

SCENARIO 28 (tell). Anne reads the letter aloud.

SCENARIO 29 (read). Bert sees that Anne reads the letter.

SCENARIO 30 (mayread). Bert orders a drink at the bar so that Anne may have read the letter.

SCENARIO 31 (bothmayread). Bert orders a drink at the bar while Anne goes to the bathroom. Both may have read the letter.

After execution of the first scenario (which is in fact a public announcement), it is common knowledge that p: in the resulting epistemic state Cp holds. This is not the case in the second scenario, but still, some common knowledge is obtained there: $C(K_1p \vee K_1\neg p)$: it is commonly known that Anne knows the content of the letter, irrespective of it being p or $\neg p$. Does this higher-order information change in Scenario 30? Yes, in this case Bert does not even know if Anne knows p or knows $\neg p$: $\neg K_2(K_1p \vee K_1\neg p)$. In Scenario 31 something similar is happening, that may best be described by saying that the agents concurrently learn that the other may have learnt p or $\neg p$. Note that in this case, both agents may have learnt p, so that p is generally known: $E_{12}p$, but they are in that case unaware of each other's knowledge, $\neg C_{12}p$, and *that* is commonly known.

Scenarios 30 and 31 are interesting, since semantically, they indicate that one cannot simply rely on the strategy of deleting states. The scenarios not only provide the agents with certainty, but also some doubts arise. After Scenario 30 for example, Bert must find an alternative state possible, in which Anne knows the contents of the letter, but also one in which Anne does not know. This is in a nutshell the main challenge in the semantics of these general updates.

Language

To a standard multi-agent epistemic language with common knowledge for a set N of agents and a set P of atoms, we add dynamic modal operators for programs that are called knowledge actions or just actions. Actions may change the knowledge of the agents involved. The formulas $\mathcal{L}_N$, the actions $\mathcal{L}_N^{\mathsf{act}}$, and the group gr of an action are defined by simultaneous induction:

DEFINITION 32 (Formulas and actions). The *formulas* $\mathcal{L}_N^{gu}(P)$ are defined by

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_n\varphi \mid C_B\varphi \mid [\alpha]\psi$$

where $p \in P$, $n \in N$, $B \subseteq N$, $\alpha \in \mathcal{L}_N^{\mathsf{act}}(P)$, and $\psi \in \mathcal{L}_{gr(\alpha)}^{gu}(P)$. The *actions* $\mathcal{L}_N^{\mathsf{act}}(P)$ are defined by

$$\alpha ::= ?\varphi \mid L_B\beta \mid (\alpha \; ! \; \alpha) \mid (\alpha \; ¡ \; \alpha) \mid (\alpha \; ; \; \beta') \mid (\alpha \cup \alpha) \mid (\alpha \cap \alpha)$$

where $\varphi \in \mathcal{L}_N^{gu}(P)$, $B \subseteq N$, $\beta \in \mathcal{L}_B^{\mathsf{act}}(P)$, and $\beta' \in \mathcal{L}_{gr(\alpha)}^{\mathsf{act}}(P)$, and where the *group* $gr(\alpha)$ of an action $\alpha \in \mathcal{L}_N^{\mathsf{act}}(P)$ is defined as: $gr(?\varphi) := \emptyset$, $gr(L_B\alpha) := B$, and $gr(\alpha \bullet \alpha') := gr(\alpha) \cap gr(\alpha')$ for $\bullet = \; ! \;, \cap, \cup, \; ; \;$. $\dashv$

The program constructor L_B is called *learning.* Action $?\varphi$ is a *test*, $(\alpha \; ; \; \alpha')$ is *sequential execution*, $(\alpha \cup \alpha')$ is *nondeterministic choice*, $(\alpha \; ! \; \alpha')$ is called *(left) local choice* and $(\alpha \; ¡ \; \alpha')$ is called *(right) local choice*, and $(\alpha \cap \alpha')$ is *concurrent execution.* The construct $L_B?\varphi$ is pronounced as 'B learn *that* φ'. Local choice $\alpha \; ! \; \alpha'$ may, somewhat inaccurately, be seen as 'from α and α', choose the first.' Local choice $\alpha \; ¡ \; \alpha'$ may be seen as 'from α and α', choose the second.' The interpretation of local choice '!' and '¡' depends on the context of learning that binds it: in $L_B(\alpha \; ! \; \alpha')$, everybody in B but not in learning operators occurring in α, α', is unaware of the specific choice for α. That choice is therefore 'local'.

EXAMPLE 33. The description in $\mathcal{L}_{12}^{\mathsf{act}}(\{p\})$ of the actions in the introduction are:

tell	$L_{12}?p \cup L_{12}?\neg p$
read	$L_{12}(L_1?p \cup L_1?\neg p)$
mayread	$L_{12}(L_1?p \cup L_1?\neg p \cup ?\top)$
bothmayread	$L_{12}(\; (L_1?p \cap L_2?p) \cup (L_1?\neg p \cap L_2?\neg p)$
	$\cup L_1?p \cup L_1?\neg p \cup L_2?p \cup L_2?\neg p \cup ?\top)$

For example, the description of read (Anne reads the letter) reads as follows: 'Anne and Bert learn that either Anne learns that she is invited for a night out in Amsterdam or that Anne learns that she has to give a lecture instead.' $\dashv$

By replacing all occurrences of '!' and '¡' in an action α by '$\cup$', except when under the scope of ?, we get the *type* $t(\alpha)$ of that action. By replacing all occurrences of '$\cup$'

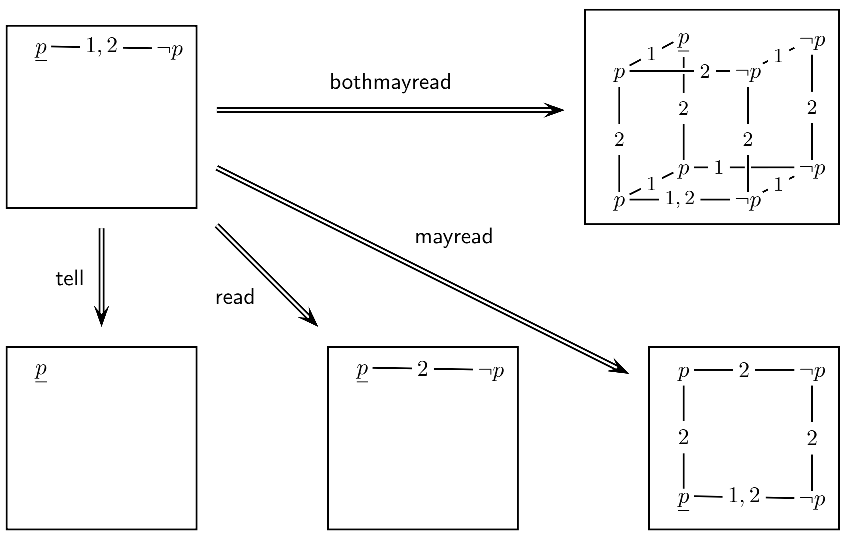


Figure 14. Epistemic states resulting from the execution of actions described in the four action scenarios. The top left figure represents (Let, u), in which it is common knowledge that both 1 and 2 are ignorant about p. For mayread and bothmayread only one of more executions is shown: namely the one in which actually nothing happens, and the one in which both 1 and 2 find out that p, respectively.

in an action α by either '!' or '¡', except when under the scope of ?, we get the set of *instances* $I(\alpha)$ of that action. Informally we write: $I(\alpha) := \{\alpha[\cup/!, ¡]\}$ If $t(\alpha) = t(\beta)$ we say that α and β are the same type of action. Furthermore, if α and β are identical modulo swapping of occurrences of '!' for '¡' or vice versa, we say that α, β are *comparable* actions. The idea here is that in the scope of an L_B operator, the agents of B know which action is executed, but the agents not in B consider all actions of the same type possible. Instead of $\alpha\ !\ \alpha'$ we also write $!\alpha \cup \alpha'$. This expresses more clearly that given choice between α and α', the agents involved in those actions choose α, whereas that choice remains invisible to the agents that learn about these alternatives but are not involved. Similarly, instead of α ¡ α' we write $\alpha\cup\ !\alpha'$.

EXAMPLE 34. The action read where Bert sees that Anne reads the letter is different from the instance of that action where *Anne is actually invited for a night out* and Bert sees that Anne reads the letter. The latter is described as $L_{12}(!L_1?p \cup L_1?\neg p)$: of the two alternatives $L_1?p$ and $L_1?\neg p$, the first is chosen, but agent 2 is unaware of that choice. The description read is its *type*. The other instance of read is $L_{12}(L_1?p\cup\ !L_1?\neg p)$. Actions $L_{12}(!L_1?p \cup L_1?\neg p)$ and $L_{12}(L_1?p\cup\ !L_1?\neg p)$ are *comparable* to each other. ⊣

Semantics and Axioms

Concerning the semantics of $\mathcal{L}_N^{gu}(P)$ (on epistemic models), we refer to Chapter 12 for the treatment of the dynamic operators, and focus here on the learning operator. Although our object language is that of [102], we focus on the semantics as explained in [7], which we coin *action model semantics*. The appealing idea in the action model semantics is that both the uncertainty about the state of the world, and that of the action taking place, are represented in two independent Kripke models. The result of performing an epistemic action in an epistemic state is then computed as a 'cross-product'. We give some more explanation by way of an example: see also Figure 15.

Model N in this figure is the model *Let*, but now we have given names s and t to the states in it. The triangular shaped model N is the *action model* that represents the knowledge and ignorance when the instance $L_{12}(L_1?p \cup L_1?\neg p \cup !\top)$ of $\mathsf{mayread}$ is carried out. The points $\mathsf{a}, \mathsf{b}, \mathsf{c}$ of the model N are also called actions, and the formulas accompanying the name of the actions are called *pre-conditions*: the condition that has to be fulfilled in order for the action to take place. Since we are in the realm of *truthful information transfer*, in order to perform an action that reveals p, the pre-condition p must be satisfied, and we write $\mathsf{pre}(\mathsf{b}) = \mathsf{p}$. For the case of nothing happening, only the precondition $\top$ need be true. Summarising, action b represents the action that agent 1 reads p in the letter, action c is the action when $\neg p$ is read, and a is for nothing happening. As with 'static' epistemic models, we omit reflexive arrows, so that N indeed represents that p or $\neg p$ is learned by 1, or that nothing happens: moreover, it is commonly known between 1 and 2 that 1 knows which action takes place, while for 2 they all look the same.

Now let $M, w = \langle W, R_1, R_2, \ldots R_m, \pi\rangle, w$ be an epistemic static state, and M, w an action in a finite action model. We want to describe what $M, w \oplus \mathsf{M}, \mathsf{w} = \langle W', R_1', R_2', \ldots R_m', \pi'\rangle, w'$, looks like — the result of 'performing' the action represented by M, w in M, w. Every action from M, w that is executable in any state $v \in W$ gives rise to a new state in W': we let $W' = \{(v, \mathsf{v}) \mid v \in W, M, v \models \mathsf{pre}(\mathsf{v})\}$. Since epistemic actions do not change any objective fact in the world, we stipulate $\pi'(v, \mathsf{v}) = \pi(v)$. Finally, when are

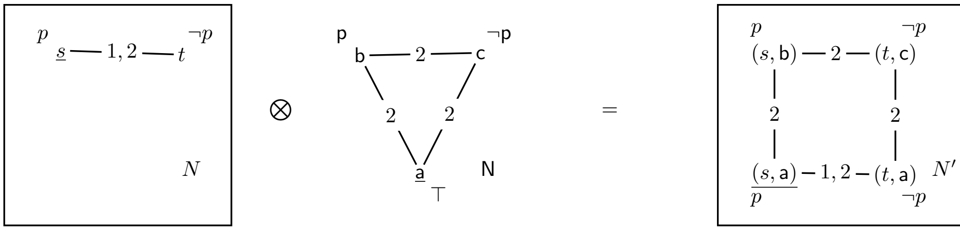


Figure 15. Multiplying the epistemic state Let, s with the action model (N, a) representing the action instance $L_{12}(L_1?p \cup L_1?\neg p \cup\, !\top)$ of $\mathsf{mayread}$

two states (v, v) and (u, u) indistinguishable for agent i? Well, he should be both unable to distinguish the originating states $(R_i uv)$, and unable to know what is happening $(\mathsf{R_i uv})$. Finally, the new state w' is of course (w, w). Note that this construction indeed gives $N, s \oplus \mathsf{N}, \mathsf{a} = N', (s, \mathsf{a})$, in our example of Figure 15. Finally, let the action α be represented by the action model state M, w. Then the truth definition under the action model semantics reads that $M, w \models [\alpha]\varphi$ iff $M, w \models \mathsf{pre}(\mathsf{w})$ implies $(M, w) \oplus (\mathsf{M}, \mathsf{w}) \models \varphi$. In our example: $N, s \models [L_{12}(L_1?p \cup L_1?\neg p \cup\, !\top)]\varphi$ iff $N', (s, \mathsf{a}) \models \varphi$.

Note that the accessibility relation in the resulting model is defined as

$$R_i(u, \mathsf{u})(v, \mathsf{v}) \;\Leftrightarrow\; R_i uv \;\&\; \mathsf{R_i uv} \tag{17}$$

As a consequence, perfect recall does not hold for $[\alpha]$: Let α be $L_{12}(L_1?p \cup L_1?\neg p \cup\, !\top)$. We then have $N, s \models K_2[\alpha]\neg(K_1 p \vee K_1 \neg p)$ (2 knows that if nothing happens, 1 will not find out whether p), but not $N, s \models [\alpha]K_2\neg(K_1 p \vee K_1 \neg p)$. We *do* have in general the following weaker form of perfect recall, however. Let M, w be a static epistemic state, and α an action, represented by some action state M, w. Let A be the set of actions that agent i cannot distinguish from M, w. Then we have

$$M, w \models \bigwedge_{\beta \in A} K_i[\beta]\varphi \rightarrow [\alpha]K_i\varphi \tag{18}$$

In words, in order for agent i to 'remember' what holds after performance of an action α, he should already now in advance that it will hold after *every epistemically possible execution* of that action. In the card example of hexa: if player 1 shows player 2 his card (which is red), than player 3 'only' knows that 2 learned that 1 holds red or white, because he cannot distinguish the action in which 1 shows red from the action in which 1 shows white. The perfect recall version (18) is a consequence of the '$\Rightarrow$'-direction of (17), the other direction gives the following generalized and conditionalised version of 'no learning': $[\alpha]K_i\varphi \rightarrow (\mathsf{pre}(\alpha) \rightarrow \bigwedge_{\beta \in A} K_i[\beta]\varphi)$. This implies that, everything that is known after a *specific* execution of an action, was already known to hold after any *indistinguishable* execution of that action.[1]

Concerning axiomatisations for dynamic epistemic logic, we provide some axioms in Table 5. These have to be added on top of $\mathbf{S}5_m$ and the usual axioms for the dynamic operators. Let us call the resulting system $\mathbf{DEL(S}5)_m$

[1] We have oversimplified the treatment of [7], in particular we have not discussed what it means that an action α is represented by an action state M, w. For further discussion, see [7], or [101], where both semantics discussed here are dealt with.

	some axioms of $\mathbf{DEL(S5)}_m$		some rules of $\mathbf{DEL(S5)}_m$
$A15$	$\langle L_B\alpha\rangle\top \leftrightarrow \mathsf{pre}(L_B\alpha)$	$R6$	$\vdash \varphi \to \psi \Rightarrow \vdash [\alpha]\varphi \to [\alpha]\psi$
$A16$	$[\alpha \,!\, \alpha']\varphi \leftrightarrow [\alpha]\varphi$	$R7$	If: for all β with $\alpha \sim_B \beta$
$A17$	$[\alpha]p \leftrightarrow (\mathsf{pre}(\alpha) \to p)$		there is a χ_β such that
$A18$	$[\alpha]\varphi \leftrightarrow \bigwedge_{\beta\in I(\alpha)}[\beta]\varphi$		(1) $\vdash \chi_\beta \to [\beta]\varphi$, and
$A19$	$\bigwedge_{\beta\sim_i\alpha} K_i[\beta]\varphi \to [\alpha]K_i\varphi$		(2) $\beta \sim_n \alpha'$ implies
$A20$	$[\alpha]K_i\varphi \to$		$\vdash (\chi_\beta \wedge \mathsf{pre}(\beta)) \to E_B\chi_{\alpha'}$
	$(\mathsf{pre}(\alpha) \to \bigwedge_{\beta\sim_i\alpha} K_i[\beta]\varphi)$		Then: $\chi_\alpha \to [\alpha]C_B\varphi$

Table 5. Epistemic axioms and rules: $i \leq m$

We elaborate shortly on the axioms: for further details we refer to [7, 102]. For any action α, a formula $\mathsf{pre}(\alpha)$ can be defined which is true exactly when α can be executed. Axiom $A15$, also called 'Learning', says that this is well-defined for learning actions, and $A17$ ('Atomic permanence') says that, like public announcement, general updates do not change the objective statements, given that the update is executable. Axiom $A16$ ('Local choice') determines the meaning of ! . Axiom $A18$ ('Action instances') formalises that the effect of an action is a combined effect of all its instances. One might have expected a distribution axiom for $[\alpha]$, but this is not sound. Such an axiom is unsound in any dynamic logic with concurrency (see Chapter 12), for the same reason: the interpretation of actions are relations between epistemic states and sets of epistemic states. The modality $[\alpha]$ corresponds to a $\forall\exists$ quantifier for which distribution does not hold. We do have a weaker form of distribution in the form of the action facilitation rule $R7$. This is all we need in the completeness proof.

Note that axioms $A19$ and $A20$ are variants of the earlier discussed principles of 'recall' and 'no learning', respectively. They give what one could call a 'compositional analysis' of pre- and post-conditions of epistemic events. Axiom $A19$ expresses that, in order to know, after α has happened, that φ, one has to know in advance that, no matter which action happens that looks like α, property φ will result. In a simple card game example: if I know after you show Ann a card that she has to know the full deal of cards among us, I should know in advance that Ann knows the deal after every card that I imagine you showing Ann. Often a contrapositive of such an axiom is appealing, in this case it reads $\langle\alpha\rangle\hat{K}_i\varphi \to \bigvee_{\beta\sim_i\alpha}\hat{K}_i\langle\beta\rangle\varphi$: if there is an execution of α after which I still consider φ a possibility, then for some action that looks the same to me as α, I imagine it possible that there is an execution which leads to φ. If you only have two cards, and you show them in a sequence to her (α), but I only see you show her a card twice, I still think it conceivable that she does not know both of your cards (φ), since I take into the consideration that you showed her twice the same card (β), after which she would not know both your cards. We leave elaboration about $A20$ to the reader.

Even the soundness of inference rule $R7$ is not easy to grasp. It mimics an induction rule where the χ formulas are used as induction hypotheses. It uses a notion of indistinguishability of actions: $\alpha \sim_B \beta$ means that group B cannot distinguish the execution of α from that of β. Also, the completeness proof of this logic is not easy, since in the canonical model one has to prove that $[\varphi]C_B\psi$ is in a maximal consistent set Γ iff for every path that runs from Γ along steps from one of the agents in B and in which φ is

true, also $[\varphi]\psi$ holds, and the right hand side of this iff has no counterpart in the object language. This motivated [90] to introduce a *relativised common knowledge operator* $C_B(\varphi, \psi)$ which exactly captures the right hand side of the mentioned equation. Both the inference rule $R7$ and the completeness proof of the logic based on this notion of common knowledge has a much more natural appearance.

8 EPISTEMIC FOUNDATIONS OF SOLUTION CONCEPTS

We now turn to the characterisation of solution concepts in games, formalised using epistemic constructs. To set the scene, let us consider the game in extensive form, depicted on the left hand side of Figure 16. It supposes we have two players, A and B, and A has to decide in the nodes labelled a, e, i and u, whereas B decides in b and d. The leaves of the tree are labelled with payoffs, the one in the left-most leaf for instance denoting that A would receive 1, and B would get 6.

A natural question now is: "suppose you are agent A. What would your decision be in the top node a?" The obvious backward induction procedure determines A's 'best' move starting from the leaves. Suppose the game would end up in node u. Since A is *rational*, he prefers an outcome of 4 over 1, and hence he would move 'left' in u; this is illustrated using the thick lines in the game on the right hand side of Figure 16. Now, B is *rational* as well, and he moreover *knows that A is rational*, so, when reaching node d, player B *knows* he has in fact a choice between a payoff of 4 (going 'left' in d) and 3 (going right, and *knowing what A will do in u*). We do the same reasoning over nodes e and i, and end up with the choices with a thick line in the figure: A would go 'right' and 'left', respectively. Again, since B is *rational and knows that A is rational*, his payoffs in node b are 4 and 3, respectively, and he will choose 'left'. Continuing in this fashion, note that A's choice for going left in a is based on (i) the fact that he is *rational*; (ii) the fact that he *knows that B is rational* and (iii) the fact that *he knows that B knows that he (A) is rational.* In short, knowledge of each other's rationality, and indeed, *common knowledge* thereof, seems to play a crucial role in rational strategic decision making.

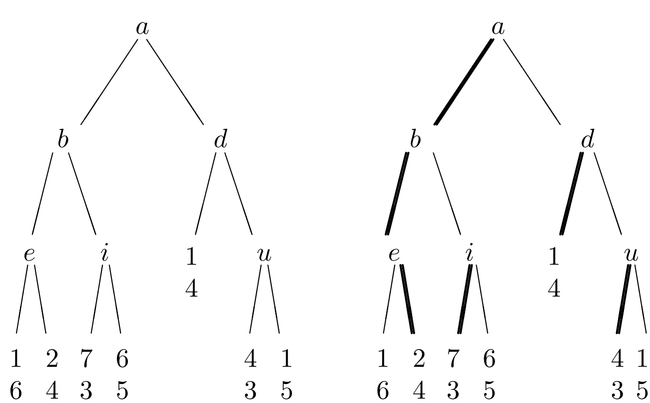


Figure 16. A Game in Extensive Form

The fact that epistemic notions are important in order to analyse certain solution

concepts in games, like common knowledge of rationality being crucial for backward induction, has been recognised for a long time, even though for certain game-theoretic solution concepts, the epistemic foundations are not always easy to determine. It seems most progress has been made in the case of strategic game forms.

8.1 Epistemic Foundations for Strategic Games

In ([18]) it is argued that a general form of epistemic characterisation results comes in a format in which one predicts the decision of each agent, given certain assumptions about each player's utility and rationality, and (iterated) knowledge thereof. The following provides an example, taken from [88]: we will mainly restrict ourselves to two player games in this section.

The proof of Theorem 7, and the backward induction algorithm applied above to recursively determine a subgame-perfect equilibrium in an extensive form game has its counterpart in strategic games. The left hand side of Table 6 represents a strategic game for two players r (choosing a row) and c (selecting a column). The entries x, y represent payoffs for player r and c, respectively. In this game, the (unique) Nash equilibrium can be achieved by iteratively removing strictly dominated strategies (see Section 2.1): since c's strategy c_3 is strictly dominated, by rationality he will not play it, so that we can remove its corresponding column from the game. Player r is rational and knows about c's rationality, and in the new game with only 2 columns that he needs to consider *he* has a dominated strategy r_3 that can be removed. Using $\mathsf{rat_i}$ to express that player i is rational, removal of r_3 is granted by the fact that $\mathsf{rat_r} \wedge \mathsf{K_r rat_c}$. Continuing in this matter, column c_2 and row r_2 can subsequently be removed, leaving us with the unique Nash equilibrium $\langle r_1, c_1 \rangle$. It seems that in our reasoning the last two steps are motivated by the fact that

$$\mathsf{rat_c} \wedge \mathsf{K_c rat_r} \wedge \mathsf{K_c K_r rat_c} \text{ and } \mathsf{rat_r} \wedge \mathsf{K_r rat_c} \wedge \mathsf{K_r K_c rat_r} \wedge \mathsf{K_r K_c K_r rat_c} \tag{19}$$

respectively. Before relating such a condition to [18]'s general format of characterisation results and his syntactic approach, we make a little detour.

$r \backslash c$	c_1	c_2	c_3
r_1	2,3	2,2	1,1
r_2	0,2	4,0	1,0
r_3	0,1	1,4	2,0

$r \backslash c$	c_1	c_2	c_3
r_1	$\mathsf{br}_r, \mathsf{br}_c$	–	–
r_2	br_c	br_r	–
r_3	–	br_c	br_r

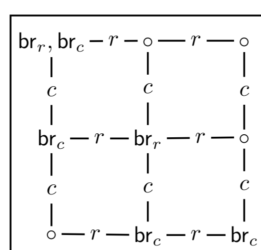


Table 6. A strategic game H for players r and c, distribution of best responses and the model M_H

A Semantic Approach

A semantic approach to clarify the nature of epistemic characterisation of solution concepts is given in [74], but here we follow a more recent approach ([88]). We do this since

the latter is closer to the Kripke models as used in this chapter, and it moreover gives a nice example of how dynamic epistemic logic (Section 7) can clarify the subtleties that are at stake here (for instance, is iterated knowledge as in (19) indeed *needed*?). Endowing finite two-player games in strategic form with an epistemic flavour uses the observation that each player knows his strategy, but not the other's. Hence, if we take the *strategy profiles* σ in a game G as the possible worlds, the epistemic indistinguishability relation that presents itself is then defined by $R_i\sigma\delta$ iff $\sigma_i = \delta_i$. This is in fact the definition that is used in the prominent *distributed* or *interpreted systems* approach to epistemic logic (see Section 5), in which global states in the overall system have a local component for each agent, each exactly knowing this local state. Thus, the *full model* over the game H of Table 6 is the left model in Figure 17; where the global states $\langle \sigma_r, \sigma_c \rangle$ are represented by their unique payoff for that profile.

As we have seen above, an algorithm to find a solution concept in a game G may transfer a model into a smaller one. Let us coin submodels of full models *general game models*. Doing so, we stay in the realm of $\mathbf{S5}_2$, since as [81] observes, every $\mathbf{S5}_2$-model is bisimular to a general game model. Now, is common knowledge of rationality *needed* to justify the elimination of dominated strategies algorithm? First of all, referring to common knowledge in full games seems to be some overkill: in every full model for two players, every pair of strategy profiles $\sigma = \langle \sigma_r, \sigma_c \rangle$ and $\tau = \langle \tau_r, \tau_c \rangle$ is connected through a third $\delta = \langle \sigma_r, \tau_c \rangle$ (and, indeed, a fourth $\gamma = \langle \tau_r, \sigma_c \rangle$), which immediately gives us $K_r K_c \varphi \rightarrow K_c K_r \varphi$ as a valid scheme on such models. This is reminiscent of the property that in interpreted systems, common knowledge in every run (roughly, every sequence of global states in the full model) is constant (see Section 5). In full models for two agents, we have $K_r K_c \varphi \rightarrow C\varphi$ (its contra-positive follows quickly from the semantic insight above: if $\neg C\varphi$ holds at σ then φ is false at some τ, which is in two steps connected through δ, giving $\hat{K}_r \hat{K}_c \neg\varphi$ —recall that $\hat{K}$ is the dual of K) so that there is a natural bound on the needed nesting of epistemic operators, and the notion of common knowledge is not really needed.

Secondly, we have to become a bit more precise of what rationality exactly amounts to in the current setting. The notion of a *best response* for agent i in a model M at state σ can be formalised as follows, where A_i is the set of actions available in the current game G to agent i. Let M be the (full or general) game model generated by G and, given a profile σ, $M, \sigma \models \pi(\cdot) \succeq_i \pi(i \mapsto a)$ means that the payoff, given profile σ is at least as big for i as for the profile that is like σ, but in which i plays a instead.

$$M, \sigma \models \mathsf{br}_i \text{ iff } M, \sigma \models \bigwedge_{a \in A_i} (\pi(i \mapsto a) \preceq_i \pi(\cdot)) \tag{20}$$

Note that a profile σ being in a Nash Equilibrium (NE) in a model M, can be described as follows: $M, \sigma \models \mathsf{NE} \leftrightarrow \bigwedge_{i \in N} \mathsf{br}_i$. In the general model M_H in the right hand side of Table 6, NE is true in the profile $\langle r_1, c_1 \rangle$. Note that we have $M_H \models (\neg K_r \mathsf{br}_r \wedge \neg K_c \mathsf{br}_c)$: neither player knows that he plays a best response, let alone that such a property is common knowledge.

Once a property br_i is true in σ in a full model M, it remains true in σ and any smaller general model M': this is because of the universal nature of br. One could alternatively define a relative notion of best response br_i^* as in (20), but with quantification restricted to all actions $A_i(M)$ in the model M. Doing so, a profile σ can *become* a best response for a player i, just because a better option for i has just been eliminated, when moving

to M'.

According to [88], a static analysis of strategic games using epistemics does not seem the right way to go: a general model for a game has players with common knowledge of rationality ($C\mathsf{Rat}$), iff Rat is true in all worlds, i.e., profiles, and such games seem not so interesting, under plausible notions of rationality. For instance, if we take rationality to mean that everybody plays a best response, then we would have that $C\mathsf{Rat}$ holds iff the model consists only of Nash equilibria. And, giving a modal logic argument, from the validity of $C(\mathsf{Rat} \rightarrow \mathsf{NE})$ one derives $C\mathsf{Rat} \rightarrow C\mathsf{NE}$, so that, assuming common knowledge of rationality, there is no need for an algorithm to eliminate worlds, since we already have $C\mathsf{NE}$: it is already common knowledge that a Nash equilibrium is played. Moreover, $C\mathsf{Rat}$ is not a plausible assumption as was illustrated in the game H of Table 6: not even any individual player knows that he is rational! The dynamic epistemic logic of Section 7 seems more appropriate to deal with assumptions about rationality.

Rationality should have an epistemic component. Rather than saying that all players play their best response, it should cover something like 'every player plays the best, given his knowledge'. We take the following notion of weak rationality wr from [88], using our notation $\hat{K}_i\varphi$ for $\exists\tau(R_i\sigma\tau \;\&\varphi$ is true at $\tau)$.

$$M, \sigma \models \mathsf{wr}_i \text{ iff } M, \sigma \models \bigwedge_{a \in Ac_i,\ a \neq \sigma_i} \hat{K}_i(\pi(\cdot) \succeq_i \pi(i \mapsto a)) \tag{21}$$

In words: a player i is weakly rational at σ, if for every alternative action for σ_i, he can imagine a profile σ', for which σ_i (which is, by the definition of the accessibility relation, σ'_i) would be at least as good for i as playing a. Alternatively: σ is a weakly rational profile for i, if for any alternative action, i does not know that σ_i is worse. It may be instructive to consider the dual reading of wr, which is $\neg \bigvee_{a \in Ac_i(M), a \neq \sigma_i} K_i(\pi(\cdot) < \pi(i \mapsto a)$: σ is weakly rational for i if there is no action a for i for which i knows that it would give him a better payoff. Yet phrased differently: σ is not weakly rational for a player, if it represents a dominated strategy. Going back to the model M_H of Figure 17, wr_c fails exactly in the third column, since there, player c *does* know to have a strictly better move than playing c_3.

THEOREM 35 ([88]). *Every finite general model has worlds where* wr_i *holds for both players* i.

Being defined as lack of knowledge, and given the fact that we are in the realm of $\mathbf{S}5_m$, we have $\mathsf{wr}_i \rightarrow K_i\mathsf{wr}_i$ as a validity, giving rationality its desired epistemic component. Moreover, we can expect players to announce that they are weakly rational, since they would know it. The next theorem uses the public announcements of Section 7.

THEOREM 36 ([88]). *Let* σ *be a strategy profile,* G *a game in strategic form, and* $M(G)$ *its associated full game model. Then the following are equivalent:*

(*i*) *Profile* σ *survives when doing iterated removal of dominated strategies;*

(*ii*) *Repeated announcement of* wr_i *stabilises at a substate* (N, σ), *for which the domain of* N *is exactly the set of states that survive when doing iterated removal of dominated strategies.*

In Figure 17, the left-most model is the model M_H from Table 6. The other models are obtained by public announcements of $\mathsf{wr}_c, \mathsf{wr}_r, \mathsf{wr}_c$ and wr_r, respectively. As a local

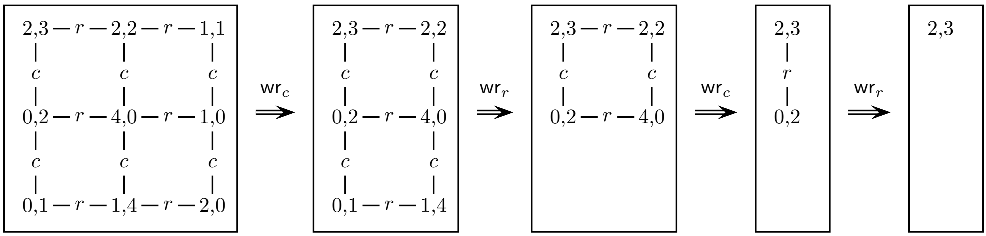


Figure 17. Epistemic model M_H for the game H (left), and after announcements of rationality.

update, this sequence can be executed in $\sigma = \langle r_1, c_1 \rangle$, but one can also conceive it as an operation on the whole model, cf. the interpretation of the learning operators in Section 7.2. Thus, in our model M_H, we have $M_H, \langle r_1, c_1 \rangle \models [\mathsf{wr}_c][\mathsf{wr}_r][\mathsf{wr}_c][\mathsf{wr}_r]C\mathit{Nash}$: if the players iteratively announce that they are weakly rational, the process of dominated strategy elimination leads them to a solution that is commonly known to be Nash.

One can give a similar analysis of *strong rationality*. A profile σ satisfies this property for a player i if i considers it possible that σ_i is as good as any other of i's actions.

$$M, \sigma \models \mathsf{sr}_i \text{ iff } M, \sigma \models \hat{K}_i \bigwedge_{a \in A_i,\ a \neq \sigma_i} (\pi(\cdot) \succeq_i \pi(i \mapsto a)) \tag{22}$$

If we abbreviate SR to be $\mathsf{sr}_r \wedge \mathsf{sr}_c$, we can again perform announcements of SR. In general, such announcements of wr, sr and SR give rise to different behaviour.

THEOREM 37 ([88]). *On full models, repeated announcements of* SR *lead to its common knowledge.*

We have assumed that games are finite. For infinite games, the reasoning put forward in this section would be naturally dealt with in some kind of fixed-point logic. We have for instance the following theorem:

THEOREM 38 ([88]). *The stable set of worlds for repeated announcement of* SR *is defined in the full game model by the greatest fixed point formula*

$$\nu p \cdot (\hat{K}_c(\mathsf{br}_c \wedge p) \wedge \hat{K}_r(\mathsf{br}_r \wedge p))$$

A Syntactic Approach

One of the main contributions of [18] is that it gives a unifying modal framework to present and relate several epistemic characterisation results of solution concepts. The claim is that such results, for a two-player strategic form game, are usually expressed in a form $\varphi(\mathsf{rat}_1, \mathsf{rat}_2, \mathsf{u}_1, \mathsf{u}_2) \rightarrow$ actions. This is demonstrated by (19), albeit that there the assumptions about the utilities (the u-propositions) are kept implicit in the model. It goes without saying that for instance player c can only apply his knowledge about r's rationality if he also knows r's utilities. A syntactic approach forces us to make such assumptions explicit.

Let us here briefly explain how [18] formalises a characterisation result for strategic form games. First of all, we need a language with knowledge operators K_i and probabilistic belief operators $\mathbf{P}_i$, where the intended interpretation of $\mathbf{P}_i(\varphi) = \mathsf{r}$ (with r a

rational number) is obvious. Basic propositions are $\mathsf{i}_1, \mathsf{i}_2, \ldots$ expressing that player i plays his first, second,... strategy. The expression $\mathsf{u}_i(k,l) = \mathsf{r}_{i,k,l}$ denotes that the utility for player i, when the strategy profile (k,l) is played, equals the number r.

The axioms needed are then $A1, A2$ and $A3$ for knowledge (see Table 2), axioms for dealing with inequalities of terms referring to probabilities of events, like $\mathsf{c} > 0 \Rightarrow (\sum_k \mathsf{q}_k \mathbf{P}_i(\varphi) \geq \mathsf{r} \Leftrightarrow \sum_k \mathsf{c}\mathsf{q}_k \mathbf{P}_i(\varphi)_k \geq \mathsf{c}\mathsf{r})$. Add the Kolmogorov axioms for $\mathbf{P}_i$, which say that $\mathbf{P}_i(\top) = 1$, $\mathbf{P}_i(\bot) = 0$, $\mathbf{P}_i(\varphi) \geq 0$, $\mathbf{P}_i(\varphi) = \mathbf{P}_i(\varphi \wedge \psi) + \mathbf{P}_i(\varphi \wedge \neg\psi)$ and $\mathbf{P}_i(\varphi) = \mathbf{P}_i(\psi)$ whenever $\varphi \leftrightarrow \psi$ is a propositional tautology. The connection axioms relate knowledge and probabilistic belief: $\mathbf{P}_i(\varphi) = 1$ is the same as $K_i\varphi$, and for every i-probability sentence φ (i.e., a Boolean combination of statements of the form $\mathbf{P}_i(\cdot) = \cdot$), we have $\varphi \rightarrow K_i\varphi$. The inference rules are $R1$ and $R2$ of Table 2.

Then there are axioms specifically for strategic game forms, called *axioms for game playing situations*. These are, respectively, that every player plays at least one strategy ($\bigwedge_i \bigvee_m \mathsf{i}_m$, where m ranges over the strategies available to player i), and not any other ($\bigwedge_i \bigwedge_{m \neq n} \neg(\mathsf{i}_m \wedge \mathsf{i}_n)$). Moreover, every player knows his chosen strategy ($\bigwedge_i \bigwedge_m K_i \mathsf{i}_m \leftrightarrow \mathsf{i}_m$), and likewise for his utilities ($\mathsf{u}_i(k,l) = \mathsf{r} \rightarrow K_i \mathsf{u}_i(k,l) = \mathsf{r}$). The scheme that captures rationality of player i is called meu_i, which is defined as the following implication:

$$K_i((\bigwedge_{k,l} \mathsf{u}_i(k,l) = \mathsf{r}_{i,k,l}) \wedge \bigwedge_l (\mathbf{P}_i(\mathsf{j}_l = \mathsf{p}_l)) \wedge \mathsf{i}_m) \rightarrow \bigwedge_k \sum_l \mathsf{p}_l \cdot \mathsf{r}_{i,m,l} \geq \sum_l \mathsf{p}_l \cdot \mathsf{r}_{i,k,l}$$

expressing that each player i aims at his Maximal Expected Utility: if i knows all his possible utilities in the game, and the probabilities with which his opponent j chooses his strategies, and i opts for his strategy m, then, for every alternative choice k for i, the expected utility for i, using i's expectations of j's behaviour, will never be bigger.

THEOREM 39 ([18]). *Let Γ be a 2-person normal form game. Assume that the following three conditions hold (where player 1 plays m and 2 plays n).*

(*i*)	*All players are rational*	$\bigwedge_i \mathsf{meu}_i$
(*ii*)	*All players know their own utility function*	$\bigwedge_i K_i \bigwedge_{k,l} \mathsf{u}_i(k,l) = \mathsf{r}_{i,k,l}$
(*iii*)	*All players know each player's actual choice*	$K_1 2_n \wedge K_2 1_m$

Then, the played action profile constitutes a Nash equilibrium, i.e., we have $\bigwedge_k \mathsf{r}_{1,m,n} \geq \mathsf{r}_{1,k,n} \wedge \bigwedge_l \mathsf{r}_{2,m,n} \geq \mathsf{r}_{2,m,l}$.

Note that the theorem above does not refer to common knowledge at all. As [18] points out, Theorem 39 is in fact well known in game theory (since 1995 [6], and even 1982, [73]). But once again, the added value of [18]'s analysis is that it can relate those approaches, and that framework for instance enables to point at weak spots in proofs of theorems similar to the one discussed here.

8.2 Epistemic Foundations for Extensive Games

The analysis in [18] is particularly interesting in the realm of extensive games, especially by pinpointing the difference between two interpretations of them: the *one-shot interpretation* on the one hand, and the *many-moment interpretation* on the other. The first interpretation is the one propagated by the key publication in game theory ([54]) and renders extensive games 'the same' as games in normal form: players act only once. Metaphorically speaking, under the one-shot interpretation, players can be thought of as making up their mind before the game really starts, and then all submit their chosen

strategy in a closed envelope to a referee. The outcome of the game is then completely determined, even without any player really 'performing' a move. In the many moment interpretation, a player only has to make a decision for the decision node he thinks he is at (and he thinks is his).

Let us briefly revisit the game on the left hand side of Figure 16. In the one-shot interpretation, as part of player 2's strategy, 2 makes a decision in node d, and rationality imposes him to choose 'left', and the decision $\langle d, left\rangle$ is part of the strategy that he will put in his envelop. (And the fact that 1 can predict this, makes the whole idea of backward induction work). In the many-moment interpretation however, if player 2 ever finds himself making a choice in node d, he obviously has to re-think his assumptions about the situation, and in particular, about player 1's rationality. It need not come as a surprise that this second interpretation has led several researchers to analyse this using *belief revision* or *counterfactuals* (see also Chapters 18 and 21 of this handbook), most notably by [75, 76].

Especially the analysis of common knowledge (or, for that matter, belief) is much harder under the second analysis, because "*... (i) at one single decision moment only the beliefs of one player are relevant ... and (ii) because we have to decide whether common beliefs involve past beliefs, or future beliefs, or both*" ([18, Chapter 4]). For a more philosophical view regarding the two interpretations, we refer to [18], here we indicate only how a formal analysis clarifies the difference.

For a proof system, we now take the axioms $A1, A2$ and $A6 - A10$ for K_i, E and C (note that since $A3$ is not assumed, we will call the attitude 'belief' but still write K_i), and the axioms for games from Section 8.1. On top of this, we add the following atoms to the language. Let i_k^x mean that player i chooses according to his k-th strategy in the subgame generated by node x, and $\mathsf{u}_i^x(k,l) = \mathsf{r}$ denotes that the utility for i, when the strategies k and l are played in the subgame generated by x, is r. An axiom KnUtSub says that these utilities are known by player i.

The principles of rationality now needed are nrat_i ("on-path rationality") and frat_i ("off-path rationality"), with the following axioms:

NRat$_{bas}$	$\mathsf{nrat}_i^x \to nsd_i(\mathcal{A}_i, \mathcal{A}_j)$
NRat$_{ind}$	$(\mathsf{nrat}_i^x \wedge K_i X_i \wedge K_i X_j) \to nsd_i(X_i, X_j)$
FRat	$\mathsf{frat}_i \leftrightarrow \bigwedge_{\rho \preceq x} \mathsf{nrat}_i^x$

The last axiom says that off-path rationality means on-path rationality at every subgame (generated by an arbitrary node x reachable form ρ, the root of the game). The statement $nsd_i(X_i, Y_j)$ means that i plays a strategy that is not strictly dominated in the game generated by i's strategies $X_i \subseteq \mathcal{A}_i$ (the latter being i's set of actions) and j's strategies $Y_j \subseteq \mathcal{A}_j$. On-path rationality is expected to model that "a player chooses a (full) strategy that given his beliefs prescribes optimal actions along the path that he expects will ensue" ([18, Chapter 4]). The base case for nrat_i then expresses that a full strategy that is optimal in a subgame, is not dominated. Similarly, the induction axiom says that a strategy that is optimal given the expectations of player i about how the game is played, is not strictly dominated in the game in which all other paths are eliminated.

THEOREM 40 ([18]). *Let Γ be a finite two-person extensive form game, with given utilities $\bigwedge_{i,k,l} \mathsf{u}_i(k,l) = \mathsf{r}_{i,k,l}$, for which the following hold*

(i)	*The players are rational*	$\bigwedge_i \mathsf{frat}_i$
(ii)	*(i) is common belief*	$C(\bigwedge_i \mathsf{frat}_i)$
(iii)	*the utilities are commonly believed*	$C(\bigwedge_{i,k,l} \mathsf{u}_i(k,l) = \mathsf{r}_{i,k,l})$

Then the players play the strategy profile that is generated using backward induction.

What about the many-moment interpretation? First of all, [18] adds some specific axioms, saying that every player knows always (that is, everywhere in the game) all payoffs, for any strategy profile, and that a player knows where he is in the game (KnWhere: $K_i^x \bigwedge_j \bigvee_{y \prec x, j_i \in D} \mathsf{j}_k^y$, where D collects all full strategies that are consistent with reaching x). The axiom KnStratM ($\mathsf{i}_k \leftrightarrow \bigwedge_{\rho \preceq x} K_i^x \mathsf{i}_k$) is rather pushing the limits of the many-moment interpretation: it says that player i, when playing strategy i_k, knows this wherever he ends up in the game.

The rationality principles *RRat* are a bit more involved, now:

$$\begin{aligned} \mathsf{rrat}_i^x \leftrightarrow \quad & ((K_i^x \textstyle\bigwedge_{k,l} \mathsf{u}_i^x(k,l) = \mathsf{r}_{i,k,l} \wedge \bigwedge_k \mathbf{P}_i^x(\mathsf{i}_k^x) = \mathsf{p}_k \wedge \bigwedge_l \mathbf{P}_j^x(\mathsf{j}_l^x) = \mathsf{p}_l \wedge \mathsf{i}_m(x)) \\ & \to \textstyle\bigwedge_k EU_i(m, x, \mathbf{P}_i^x) \geq EU_i(k, x, \mathbf{P}_i^x)) \end{aligned}$$

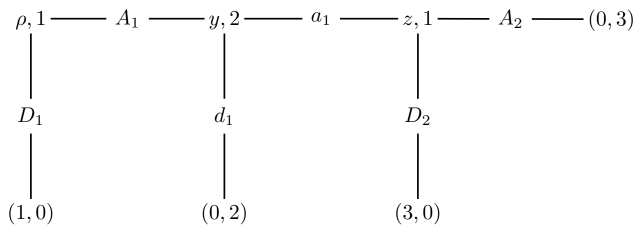
The property *RRat* above states that it holds for player i at node x iff the following implication holds. The antecedent states that i knows at x the utilities for every strategy profile, and he has some probabilistic beliefs about the strategies that both he and his opponent j will choose at x. Moreover, i in fact chooses the action prescribed by his m-th strategy at x. If this is fulfilled, then the chosen strategy m should give at least the utility obtained using any other option k. Here, $EU_i(m, x, \mathbf{P}_i^x)$ is the expected utility for i in the node y that he reaches immediately from x when using strategy m.

We now show how a negative result can be proven 'during progress in a game', using the two-player centepede example. In [70]'s variant of centepede, (see Figure 18), there are three decision nodes ρ, y and z. Of the two players, player 2 only moves in y. The payoffs for the players are represented as pairs (u_1, u_2) in the leaves of the tree, where u_i represents player i's payoff. We leave it to the reader to check that backward induction leads player 1 to play down D_1 at the root ρ. [18] formalises [70]'s claim about this game that says that in node y, there cannot be common knowledge of rationality.

To make the proof work, we need two persistence properties. The first applies to strategies, and says that i's prediction about j's choices should not change if nothing unexpected happens: if the nodes x, y and z appear in a path on the tree (in that order), then $\mathbf{P}_i^x(\mathsf{j}_j^z)$ and $\mathbf{P}_i^y(\mathsf{j}_j^z)$ should coincide. The other preservation property applies to rationality, and says that i will not change his rationality assumptions about j during a period that j did not make a move: $K_i^x \mathsf{rrat}_j^y \leftrightarrow K_i^x \mathsf{rrat}_j^z$ if x, y and z appear in a path on a tree (in that order, possibly $x = y$), and z is the first node from y where j moves.

The proof in [18] of the claim $C^y \mathsf{rrat}^y \to \perp$ runs as follows. First of all, we show $(a) C^y \mathsf{rrat}^y \to K_2^y K_1^\rho d_1$ (if the principle of rrat at y is common knowledge at y, then 2 knows at y that 1 knows at ρ that 2 plays d_1). To prove it, we prove $(i) C^y \mathsf{rrat}^y \to K_1^\rho d_1$ and then apply necessitation for K_2^y. Using the preservation of strategies principle, it is sufficient to prove $(ii) C^y \mathsf{rrat}^y \to K_1^y d_1$. Applying K_1^y-necessitation to the rationality axiom gives us $K_1^y \mathsf{rrat}_2^y \wedge K_1^y K_2^y D_2 \to K_1^y d_1$, so that, to prove (ii), we are done if we show $(iii) C^y \mathsf{rrat}^y \to (K_1^y \mathsf{rrat}_2^y \wedge K_1^y K_2^y D_2)$. The first conjunct is immediate, for the second, use the persistence of rationality (for K_2^y) and necessitation (for K_1^y) to show $K_1^y K_2^y \mathsf{rrat}_1^y \to K_1^y K_2^y \mathsf{rrat}_1^z$. Applying rationality to this gives the second conjunct.

Furthermore, from the persistence of rationality, it is easy to show $(b) C^y \mathsf{rrat}^y \to$



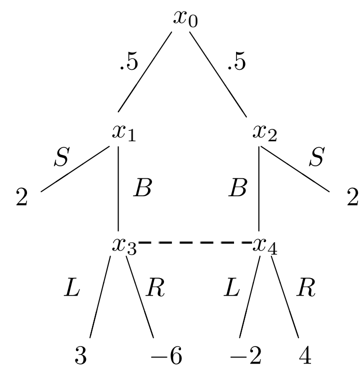


Figure 18. Centepede (left) and a one-person game with perfect recall (right)

$K_2^y \mathsf{rrat}_1^\rho$. Also, rationality and necessitation give $(c)(K_2^y K_1^\rho d_1 \wedge K_2^y \mathsf{rrat}_1^\rho) \rightarrow K_2^y \neg A_1$. Combining $(a), (b)$ and (c) immediately gives us $C^y \mathsf{rrat}^y \rightarrow K_2^y \neg A_1$. However, the D axiom $(\neg K_2^y \bot)$ together with the derivability of $K_2^y A_1$ from the KnWhere axiom (see above) then establishes our claim $C^y \mathsf{rrat}^y \rightarrow \bot$.

Which would again emphasise the intuition uttered by many game theorists that in the analysis with which we started this section, backward induction must be based on some form of counterfactual reasoning.

8.3 On the Representation of Games with Imperfect Information

The manuscript [28] gives a neat discussion on issues that arise when modelling a game with imperfect information. We already argued that the extensive form of a game gives a more natural account of the temporal aspects of a game than its strategic form. For imperfect information games, we can add equivalence relations to denote the player's information sets. However, such information sets can for instance not capture knowledge or beliefs that a player has over the strategies of the other players, or about their rationality. There are of course other ways of representing uncertainty in games, like for instance what [28] calls a *state space* representation, in which one for instance associates a strategy profile to every state (cf. the representation used in Section 8.1 and Figure 17). But in those representations, one loses the ability again to represent the temporal information. In fact, [28]'s main point is a plea to use *interpreted systems* (see Section 5) to model games of imperfect information, because they make explicit where the knowledge comes from.

Let us present [28]'s argument representing games as systems using the game that it borrows from [67], represented as the game on the right of Figure 18. In x_0, nature choses randomly either x_1 or x_2, and then the only player can chose either S or B. He cannot distinguish x_3 from x_4, where he has the option L or R. Using uniform strategies only, let the strategy $\sigma = \langle B, S, R \rangle$ denote that our player plays B in x_1, S in x_2 and R in the information set $\{x_3, x_4\}$. Similarly, let σ' be $\langle B, S, L \rangle$. One easily verifies that the payoff 3 when playing σ is the maximal expected utility: no other strategy is as good. One could easily implement such a strategy using an interpreted system where the local state ℓ of the player would indicate whether he is in $\{x_1\}$, in $\{x_2\}$ or in $\{x_3, x_4\}$.

Now [67] argues that if node x_1 is reached, the agent is better off by changing from strategy σ to σ'. And, as [28] argues, this is right, if the agent is able to remember that he switched strategies. Since the agent cannot distinguish x_3 from x_4, he should use a uniform strategy and do the same in both. However, if the agent would have perfect recall, he would distinguish the nodes, and by allowing him to remember that he switched strategies he can simulate perfect recall: if he ends up in $\{x_3, x_4\}$, he must realize that he came through x_1 and is now playing σ'. Under this assumption it is not clear anymore what it means to have a dotted line between x_3 and x_4, since the states become distinguishable. Using interpreted systems it is very natural to encode in the local state of the agent that he switched strategies, by recording the strategies he has been playing until now, for instance.

9 GAME LOGIC

Computer science has developed many different logics for reasoning about the behaviour of computer programs or algorithms. Propositional Dynamic Logic (PDL) (see [33, 34] and also Chapter 12 of this handbook) is a well-studied example which contains expressions $[\pi]\varphi$ stating that all terminating executions of program π will end in a state satisfying φ. What distinguishes PDL from simple multi-modal logic is that π can be a complex program such as $p_1; p_2$, the sequential execution of first p_1 and then p_2. PDL formalises properties of programs in its axioms such as $[p_1; p_2]\varphi \leftrightarrow [p_1][p_2]\varphi$ which completely characterises the sequential composition operator.

Nondeterministic programs may be viewed as particularly simple games, namely 1-player games. Examples of algorithms involving more than one player or agent are cake-cutting algorithms, voting procedures and auctions. Also structurally, games are very similar to programs. One game may be played after another, a player may choose to play a game repeatedly, and so on. Hence, one may expect reasoning about games to be similar to reasoning about programs, and consequently game logics should resemble program logics.

Game Logic (GL), introduced in [58, 59], is a generalisation of PDL for reasoning about determined 2-player games, allowing us to describe algorithms like the cake-cutting algorithm and to reason about their correctness. GL extends PDL by generalising its semantics and adding a new operator to the language. The meta-theoretic study of PDL has given us valuable insights, e.g., into the complexity of reasoning about programs and the expressive power of various programming constructs. By comparing GL to PDL, we can get an idea of how reasoning about games differs from reasoning about programs. In this section, we can only introduce the syntax and semantics of Game Logic (Subsection 9.1) and discuss some of its central meta-theoretic properties (Section 9.2). Further topics of research are briefly mentioned in Section 9.3. The interested reader is referred to a recent survey article on the subject [65] with more detailed references.

9.1 *Syntax and Semantics*

The games of Game Logic involve two players, player 1 (Angel) and player 2 (Demon). Just like PDL, the language of GL consists of two sorts, propositions and (in the case of GL) games. Given a set of atomic games Γ_0 and a set of atomic propositions Φ_0, games γ and propositions φ can have the following syntactic forms, yielding the set of GL-games

Γ and the set of GL-propositions/formulas Φ:

$$\begin{array}{ll} \gamma := & g \mid \varphi? \mid \gamma_1;\gamma_2 \mid \gamma_1 \cup \gamma_2 \mid \gamma^* \mid \gamma^d \\ \varphi := & \bot \mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \langle\gamma\rangle\varphi \end{array}$$

where $p \in \Phi_0$ and $g \in \Gamma_0$. As usual, we define $[\gamma]\varphi := \neg\langle\gamma\rangle\neg\varphi$. The formula $\langle\gamma\rangle\varphi$ expresses that Angel has a strategy in game γ which ensures that the game ends in a state satisfying φ. $[\gamma]\varphi$ expresses that Angel does not have a $\neg\varphi$-strategy, which by determinacy is equivalent to saying that Demon has a φ-strategy. To provide some first intuition regarding the game operations, $\gamma_1 \cup \gamma_2$ denotes the game where Angel chooses which of the two subgames to continue playing, and the sequential composition $\gamma_1;\gamma_2$ of two games consists of first playing γ_1 and then γ_2. In the iterated game γ^*, Angel can choose how often to play γ (possibly not at all); each time she has played γ, she can decide whether to play it again or not. Playing the dual game γ^d is the same as playing γ with the players' roles reversed, i.e., any choice made by Angel in γ will be made by Demon in γ^d and vice versa. The test game $\varphi?$ consists of checking whether a proposition φ holds at that position. This construction can be used to define conditional games such as $(p?;\gamma_1) \cup (\neg p?;\gamma_2)$: Suppose for instance that p holds at the present state of the game, then Angel will naturally choose the left side (if she chooses the right side, she loses at once), and γ_1 will be played. Otherwise Angel will choose right and γ_2 will be played.

Thanks to the dual operator, demonic analogues of these game operations can be defined. Demonic choice between γ_1 and γ_2 is denoted as $\gamma_1 \cap \gamma_2$ which abbreviates $(\gamma_1^d \cup \gamma_2^d)^d$. Demonic iteration of γ is denoted as $\gamma^\times$ which abbreviates $((\gamma^d)^*)^d$.

A further note on iteration: In γ^*, it is essential that Angel can decide as the game proceeds whether to continue playing another round of γ or not. The game where Angel has to decide up front how often to play γ is a different game and in general more difficult for Angel to win than γ^*. For programs (i.e., in PDL), these two notions of iteration coincide, but for games (i.e., in GL), they do not.

The formal semantics of Game Logic utilises the following game models which generalise the neighbourhood models or minimal models used in the semantics of non-normal modal logics [17]. A *game model* $\mathcal{M} = ((S, \{E_g | g \in \Gamma_0\}), V)$, consists of a set of states S, a valuation $V : \Phi_0 \to \mathcal{P}(S)$ for the propositional letters and a collection of *neighbourhood functions* $E_g : S \to \mathcal{P}(\mathcal{P}(S))$ which are monotonic, i.e. $X \in E_g(s)$ and $X \subseteq X'$ imply $X' \in E_g(s)$.

The intuitive idea is that $X \in E_g(s)$ (alternative notation: sE_gX) holds whenever Angel has a strategy in game g to achieve X. Intuitively, neighbourhood functions are reduced monotonic effectivity functions (see Section 2.3) in that they only represent the effectivity of a single player. Since we are dealing with determined games only, it is sufficient to represent Angel's effectivity. Then Demon is effective for X if and only if Angel is not effective for its complement $\overline{X}$. Furthermore, neighbourhood functions do not satisfy the two conditions we put on effectivity functions, i.e., due to the presence of the test operator ?, we allow that $\emptyset \in E_g(s)$ and $S \notin E_g(s)$. For Angel, these conditions correspond to heaven (a game where Angel can achieve anything at all) and hell (a game where Angel can achieve nothing whatsoever).

By simultaneous induction, we define truth in a game model on the one hand and the neighbourhood functions for non-atomic games on the other hand. The function $E_\gamma : S \to \mathcal{P}(\mathcal{P}(S))$ is defined inductively for non-atomic games γ (where $E_\gamma(Y) = \{s \in$

$S|sE_\gamma Y\}$), and the truth of a formula φ in a model $\mathcal{M} = ((S, \{E_g | g \in \Gamma_0\}), V)$ at a state s (denoted as $\mathcal{M}, s \models \varphi$) is defined by induction on φ. We define

$$\begin{array}{lll}
\mathcal{M}, s \not\models \bot & & \\
\mathcal{M}, s \models p & \text{iff} & p \in \Phi_0 \text{ and } s \in V(p) \\
\mathcal{M}, s \models \neg\varphi & \text{iff} & \mathcal{M}, s \not\models \varphi \\
\mathcal{M}, s \models \varphi \vee \psi & \text{iff} & \mathcal{M}, s \models \varphi \text{ or } \mathcal{M}, s \models \psi \\
\mathcal{M}, s \models \langle\gamma\rangle\varphi & \text{iff} & sE_\gamma\varphi^{\mathcal{M}}
\end{array}
\qquad
\begin{array}{rl}
E_{\alpha;\beta}(Y) & = E_\alpha(E_\beta(Y)) \\
E_{\alpha\cup\beta}(Y) & = E_\alpha(Y) \cup E_\beta(Y) \\
E_{\varphi?}(Y) & = \varphi^{\mathcal{M}} \cap Y \\
E_{\alpha^d}(Y) & = \overline{E_\alpha(\overline{Y})} \\
E_{\alpha^*}(Y) & = \bigcap\{X \subseteq S | Y \cup E_\alpha(X) \subseteq X\}
\end{array}$$

where $\varphi^{\mathcal{M}} = \{s \in S | \mathcal{M}, s \models \varphi\}$. For iteration, our definition yields a least fixpoint, i.e., the smallest set $X \subseteq S$ such that $Y \cup E_\alpha(X) = X$.

9.2 Metatheory

Axiomatisation and Expressiveness

PDL has a very natural complete axiomatisation, and given the similarity between programs and games, one can hope that adding an axiom for the dual operator is all that is needed to obtain a complete axiomatisation of GL. A small problem is presented by the induction principle

$$(\varphi \wedge [\gamma^*](\varphi \to [\gamma]\varphi)) \to [\gamma^*]\varphi$$

which is valid in PDL but invalid in GL. While this induction principle usually forms part of the axiomatic basis of PDL, alternative axiomatisations exist which instead of the induction axiom use a fixpoint inference rule (given below). It turns out that this rule is indeed sound, yielding the following axiomatic system.

Let **GL** be the smallest set of formulas which contains all propositional tautologies together with all instances of the axiom schemas of Figure 19, and which is closed under the rules of Modus Ponens, Monotonicity and Fixpoint, also presented in Figure 19.

Axioms:

$$\langle\alpha \cup \beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\varphi \vee \langle\beta\rangle\varphi$$
$$\langle\alpha; \beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\langle\beta\rangle\varphi$$
$$\langle\psi?\rangle\varphi \leftrightarrow (\psi \wedge \varphi)$$
$$(\varphi \vee \langle\gamma\rangle\langle\gamma^*\rangle\varphi) \to \langle\gamma^*\rangle\varphi$$
$$\langle\gamma^d\rangle\varphi \leftrightarrow \neg\langle\gamma\rangle\neg\varphi$$

Inference Rules:

$$\frac{\varphi \quad \varphi \to \psi}{\psi} \qquad \frac{\varphi \to \psi}{\langle\gamma\rangle\varphi \to \langle\gamma\rangle\psi} \qquad \frac{(\varphi \vee \langle\gamma\rangle\psi) \to \psi}{\langle\gamma^*\rangle\varphi \to \psi}$$

Figure 19. The axioms and inference rules (Modus Ponens, Monotonicity and Fixpoint Rule) of Game Logic.

Intuitively, the axiom for iteration (our fourth axiom) states that $\langle\gamma^*\rangle\varphi$ is a pre-fixpoint of the operation $\varphi \vee \langle\gamma\rangle X$. Conversely, the fixpoint rule states that $\langle\gamma^*\rangle\varphi$ is the *least* such pre-fixpoint.

THEOREM 41. **GL** *is sound with respect to the class of all game models.*

At the time of writing, completeness of **GL** is still open, but some weaker results exist. If x is an operator of Game Logic such as d or $*$, let $\mathbf{GL}^{-x}$ denote Game Logic without the operator x, i.e., restricted to formulas without the operator and without the axioms involving it.

THEOREM 42 ([59, 62]). *Dual-free Game Logic* $\mathbf{GL}^{-d}$ *and iteration-free Game Logic* $\mathbf{GL}^{-*}$ *are both sound and complete with respect to the class of all game models.*

Hence, we have axiomatic completeness for $\mathbf{GL}^{-d}$ as well as $\mathbf{GL}^{-*}$, but iteration together with duality remains a problem for axiomatisation. It may then not come as a surprise that it is precisely this combination which gives Game Logic its expressive power. This is most easily demonstrated when considering GL interpreted over Kripke models.

Kripke models can be viewed as special kinds of game models, namely game models where neighbourhood functions have a special property called *disjunctivity*: for every $g \in \Gamma_0$ and $V \subseteq \mathcal{P}(S)$ we have $\bigcup_{X \in V} E_g(X) = E_g(\bigcup_{X \in V} X)$. Hence, one may also investigate Game Logic when interpreted over Kripke models (i.e., disjunctive game models) only. Dual-free Game Logic over Kripke models is nothing but Propositional Dynamic Logic. Since the absence of infinite g-branches is not expressible in PDL but can be expressed by the GL-formula $\langle (g^d)^* \rangle \bot$, Game Logic over Kripke models is strictly more expressive than PDL.

Over Kripke models, full Game Logic can be embedded into the modal μ-calculus (see Chapter 12 and [41]). In fact, the 2-variable fragment of the μ-calculus suffices for the embedding, and since it has been shown that the variable hierarchy of the μ-calculus is strict [10], Game Logic is a proper fragment of the μ-calculus.

Complexity

The two central complexity measures associated with a logic are the complexity of model checking and the complexity of the satisfiability problem. For the latter, we are interested to know the difficulty of determining whether a formula φ is satisfiable, measured in the length of the formula $|\varphi|$.

Using a translation of Game Logic formulas into modal μ-calculus formulas, we can reduce Game Logic satisfiability to μ-calculus satisfiability. Since Game Logic and the standard modal μ-calculus are interpreted over different models, the models have to be translated as well. The result obtained via this procedure is the following:

THEOREM 43 ([59, 62]). *The complexity of the satisfiability problem for Game Logic is in EXPTIME.*

Turning now toward model checking, given a game model $\mathcal{M}$ and a Game Logic formula φ, we want to determine the set of states s for which $\mathcal{M}, s \models \varphi$ holds. The complexity of model checking is usually measured in terms of the size of the formula and the size of the model. Given a game model $\mathcal{M} = (S, \{E_g | g \in \Gamma_0\}, V)$, we define its size $|\mathcal{M}|$ as

$$|\mathcal{M}| = |S| + \sum_{\{s|s \in S\}} \sum_{\{g|g \in \Gamma_0\}} \sum_{\{X|sE_gX\}} |X|$$

Note that in practice one will want to represent game models more succinctly by only representing the non-monotonic core of E_g, i.e., we will disregard all those triples (s, g, X) for

which there is some $Y \subseteq X$ such that sE_gY. While in some cases (e.g., in case the game model corresponds to a Kripke model) such a representation can yield a dramatically more efficient representation, in general this is not the case, and hence the complexity result below cannot be essentially improved by disregarding supersets.

In the modal μ-calculus, the complexity of current model-checking algorithms depends on the alternation depth of a formula, i.e., on the nestings of least and greatest fixpoint operators in the formula. For GL, the situation is similar, since angelic iteration corresponds to a least fixpoint and demonic iteration to a greatest fixpoint. Hence, the maximal number of nested demonic and angelic iterations will determine the model-checking complexity of the formula. As an example, the alternation depth of a formula φ, denoted as $ad(\varphi)$ will be higher for $\langle (g^*)^\times \rangle p$ than for $\langle (g^*)^* \rangle p$.

THEOREM 44 ([62]). *Given a Game Logic formula φ and a finite game model $\mathcal{M}$, model checking can be done in time $O(|\mathcal{M}|^{ad(\varphi)+1} \times |\varphi|)$.*

9.3 Other Topics

The notion of bisimulation has been the central notion of process equivalence for modal and dynamic logic (see Chapter 1, 5 and [12]). As for modal logic, modal formulas are invariant for bisimulation, i.e., bisimular states satisfy the same modal formulas, and for finite models, the converse holds as well. Furthermore, it has been shown that the bisimular fragment of first-order logic is precisely modal logic [80].

Bisimulation can be generalised from Kripke models to game models. Two states s and s' are bisimular in case (i) they satisfy the same atomic properties, (ii) if player 1 has an X-strategy in game g from s, she also has an X'-strategy in g from s', where every state in X' must have a bisimular state in X, and (iii) analogously for strategies from s'. Intuitively, bisimular states cannot be distinguished by either their atomic properties or by playing atomic games. It can be shown that this notion of bisimulation generalises bisimulation as it is normally defined for Kripke models. Similarly, one can show that bisimular game models satisfy the same GL formulas, and one can even partially characterise the game operations of Game Logic in terms of bisimulation [61].

The operations of Game Logic have also been studied from an algebraic perspective [87]. We call two game expressions γ_1 and γ_2 *equivalent* provided that $E_{\gamma_1} = E_{\gamma_2}$ holds for all game models. Put differently, γ_1 and γ_2 are equivalent iff $\langle\gamma_1\rangle p \leftrightarrow \langle\gamma_2\rangle p$ is valid for a p which occurs neither in γ_1 nor in γ_2. When γ_1 and γ_2 are equivalent, we say that $\gamma_1 = \gamma_2$ is a *valid game identity.*

As an example, if a choice of a player between x and y is followed by game z in any case, then the player might as well choose between $x; z$ and $y; z$ directly. Hence, $(x \cup y); z = x; z \cup y; z$ is a valid game identity. The right-distributive law $x; (y \cup z) = x; y \cup x; z$ on the other hand is not valid. In the first game, player 1 can postpone her choice until after game x has been played. She may have a winning strategy which depends on how x is played, and hence such a strategy will not necessarily be winning in the second game, where she has to choose before x is played.

Game algebra further illustrates the link between games and processes that we already discussed in section 3.3. Basic game algebra studies the game operations of sequential composition, choice (demonic and angelic) and duality. The test-operator is excluded since it would take us out of the purely algebraic framework. The central result obtained for basic game algebra is a complete axiomatisation of the set of valid game identities

[25, 105]. So far, the complete axiomatisation has not been extended to a version of game algebra which includes iteration.

10 COALITION LOGIC

Modal logic describes transition systems at a very abstract level. The transition relation does not specify what or who is involved in making the transition, it only models all the possible evolutions of the system. Game forms on the other hand explicitly represent how different agents can contribute to the system's evolution by modeling the agents' strategic powers.

The semantic models of Coalition Logic [63, 64] make use of strategic games to describe the agents' abilities to influence system transitions. Using α-effectivity functions (see Section 2.3), we can formalise an agent's ability to bring about φ. More generally, the expression $[C]\varphi$ states that the coalition C, a subset of agents/players, can bring about φ. After presenting the syntax and semantics of Coalition Logic, Section 10.2 presents an axiomatisation of coalitional ability in extensive games of almost perfect information. Complexity results concerning the satisfiability problem will be discussed in Section 10.3.

10.1 Syntax and Semantics

Assuming a finite nonempty set of agents or players N, we define the syntax of Coalition Logic as follows. Given a set of atomic propositions Φ_0, a formula φ can have the following syntactic form:

$$\varphi := \quad \bot \mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid [C]\varphi$$

where $p \in \Phi_0$ and $C \subseteq N$. We define $\top$, $\wedge$, $\rightarrow$ and $\leftrightarrow$ as usual. In case $C = \{i\}$, we write $[i]\varphi$ instead of $[\{i\}]\varphi$.

A *coalition frame* is a pair $\mathcal{F} = (S, G)$ where S is a nonempty set of states (the universe) and G assigns to every state $s \in S$ a strategic game form $G(s) = (N, (\Sigma_i)_{i\in N}, o, S)$. At state s, the game form $G(s)$ represents the possible transitions based on the strategic choices of the players. A *coalition model* is a pair $\mathcal{M} = (\mathcal{F}, V)$ where $\mathcal{F}$ is a coalition frame and $V : \Phi_0 \rightarrow \mathcal{P}(S)$ is the usual valuation function for the propositional letters. Given such a model, truth of a formula in a model at a state is defined as follows:

$$\begin{array}{ll}
\mathcal{M}, s \not\models \bot & \\
\mathcal{M}, s \models p & \text{iff } p \in \Phi_0 \text{ and } s \in V(p) \\
\mathcal{M}, s \models \neg\varphi & \text{iff } \mathcal{M}, s \not\models \varphi \\
\mathcal{M}, s \models \varphi \vee \psi & \text{iff } \mathcal{M}, s \models \varphi \text{ or } \mathcal{M}, s \models \psi \\
\mathcal{M}, s \models [C]\varphi & \text{iff } \varphi^{\mathcal{M}} \in E^{\alpha}_{G(s)}(C)
\end{array}$$

where $\varphi^{\mathcal{M}} = \{s \in S | \mathcal{M}, s \models \varphi\}$. Hence, a formula $[C]\varphi$ holds at a state s iff coalition C is α-effective for $\varphi^{\mathcal{M}}$ in $G(s)$.

Coalition frames are essentially extensive game forms of *almost perfect* information. The only source of imperfect information is that players make choices simultaneously. After the choices are made at state s, a new state t results and the choices become common knowledge, before new (and possibly different) choices can be made. Note that coalition frames are game graphs rather than trees (though they can be unravelled into trees), and they contain no terminal states since every state is associated to a game.

It is possible to isolate (semantically and axiomatically) the class of coalition frames corresponding to extensive game forms of *perfect* information, see [63] for details.

Note that by Theorem 12, we can equivalently view a coalition frame $\mathcal{F}$ as a pair(S, E), where

$$E : S \to (\mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S)))$$

assigns to every state $s \in S$ a monotonic, N-maximal and superadditive effectivity function $E(s)$ (see Section 2.3). Using this formulation, we can then simply define $\mathcal{M}, s \models [C]\varphi$ iff $\varphi^{\mathcal{M}} \in E(s)(C)$. From a logical point of view, this second formulation directly in terms of effectivity functions is preferable. It simplifies meta-theoretic reasoning, e.g. by immediately suggesting certain axioms of coalitional power, and it also demonstrates that coalition models are essentially neighbourhood models, providing a neighbourhood relation for every coalition of players. Neighbourhood models have been the standard semantic tool to investigate non-normal modal logics (see, e.g. [17]), and techniques used to provide complete axiomatisations for such logics can also be adapted to Coalition Logic.

The two extreme coalitions $\emptyset$ and N are of special interest. $[N]p$ expresses that some possible next state satisfies p, whereas $[\emptyset]p$ holds if no agent needs to do anything for p to hold in the next state. Hence, $[N]p$ corresponds to $\Diamond p$ in standard modal logic whereas $[\emptyset]p$ corresponds to $\Box p$. If $|N| = 1$, e.g. $N = \{1\}$, coalition models are just serial Kripke models, i.e., Kripke models where every state has at least one successor. In this case, $[\emptyset]\varphi$ coincides with $\Box\varphi$ and $[1]\varphi$ with $\Diamond\varphi$.

10.2 Axiomatics

Let $\mathbf{CL_N}$ denote the smallest set of formulas which contains all propositional tautologies together with all instances of the axiom schemas listed in Figure 20, and which is closed under the rules of Modus Ponens and Equivalence given below:

$$\frac{\varphi \quad \varphi \to \psi}{\psi} \qquad\qquad \frac{\varphi \leftrightarrow \psi}{[C]\varphi \leftrightarrow [C]\psi}$$

Note that axioms $(\bot)$ and $(\top)$ correspond to the two basic assumptions we made for effectivity functions in Definition 9. The remaining three axioms express the conditions of Theorem 12, N-maximality, monotonicity and superadditivity.

$(\bot)$	$\neg[C]\bot$
$(\top)$	$[C]\top$
(N)	$\neg[\emptyset]\neg\varphi \to [N]\varphi$
(M)	$[C](\varphi \wedge \psi) \to [C]\psi$
(S)	$([C_1]\varphi_1 \wedge [C_2]\varphi_2) \to [C_1 \cup C_2](\varphi_1 \wedge \varphi_2)$
	where $C_1 \cap C_2 = \emptyset$

Figure 20. The axiom schemas of Coalition Logic

THEOREM 45 ([63]). $\mathbf{CL_N}$ *is sound and complete with respect to the class of all coalition models.*

KD is the normal modal logic for reasoning about serial Kripke models. In the formulation closest to Coalition Logic, **KD** is the set of formulas containing all propositional tautologies, closed under the rules of Modus Ponens and Equivalence (for $\Box$ only), and containing the axioms of Figure 21.

$\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$	$\Box(\varphi \wedge \psi) \leftrightarrow (\Box\varphi \wedge \Box\psi)$
$\Box\top$	$\Diamond\top$

Figure 21. Axioms of **KD**

The following result states that **KD** is precisely single-agent Coalition Logic. The result is the axiomatic analogue to our earlier observation that if $|N| = 1$, coalition models are simply serial Kripke models.

THEOREM 46 ([63]). *Identifying* $[\emptyset]\varphi$ *with* $\Box\varphi$ *and* $[1]\varphi$ *with* $\Diamond\varphi$*, we have* $\mathbf{KD} = \mathbf{CL}_{\{\mathbf{1}\}}$.

The logic $\mathbf{CL_N}$ is the most general and hence weakest coalition logic which has been investigated. The only assumption made is that at every state, the coalitional power distribution arises from a situation which can be modeled as a strategic game. Additional axioms can be added for characterising special kinds of strategic interaction. For example, in order to characterise extensive game forms of perfect information, one adds the axiom

$$[N]\varphi \to \bigvee_{i\in N} [i]\varphi,$$

expressing that everything which can be achieved at all can be achieved already by some individual. This axiom will enforce that at every state there is a single agent who can determine the next state independent of the other agents. Note that the converse implication can be derived in $\mathbf{CL_N}$.

Nash-consistent Coalition Logic [32] is another example of a logic stronger than $\mathbf{CL_N}$. By adding a further axiom to $\mathbf{CL_N}$, one can characterise the class of Nash-consistent coalition models, i.e., coalition models where the strategic game form associated to every state must have a Nash equilibrium under every possible preference profile. Nash-consistent models can be viewed as stable systems, in the sense that no matter what the agents' preferences are, there is a stable strategy profile, a profile for which individual deviation is irrational.

10.3 Complexity

We assume in our discussion of complexity that $|N| > 1$. As was mentioned, the two extreme coalitions $\emptyset$ and N allow one to capture necessity and possibility. For this reason, the normal modal logic **KD** forms a fragment of Coalition Logic, thereby establishing a PSPACE lower bound for the basic Coalition Logic over general coalition models. As it turns out, this bound is tight, i.e., we have the following result.

THEOREM 47 ([63]). *The complexity of the satisfiability problem for Coalition Logic is PSPACE-complete.*

Via the satisfiability problem, we can compare the complexity of reasoning about games of various different kinds. For instance, it turns out that restricting the class

of coalition models to perfect information models, the satisfiability problem remains PSPACE-complete. Hence, given a coalitional specification, finding an extensive game (form) of almost perfect information satisfying the specification is not harder nor simpler than finding an extensive game of perfect information.

Besides comparing reasoning over different classes of games, we can compare reasoning about groups to reasoning about individuals. Let the *individual fragment of Coalition Logic* be the set of formulas of Coalition Logic where all modalities only involve singleton coalitions, i.e.

$$\varphi := \quad \bot \mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid [i]\varphi$$

where $p \in \Phi_0$ and $i \in N$. The individual fragment is strictly less expressive than full Coalition Logic, since the formula $[C]p$ is in general not equivalent to any formula involving only singleton coalitions. More precisely, there is no formula φ of the individual fragment such that $\varphi^{\mathcal{M}} = [C]p^{\mathcal{M}}$ for every coalition model $\mathcal{M}$.

THEOREM 48 ([64]). *The complexity of the satisfiability problem for the individual fragment of Coalition Logic is NP-complete.*

Hence, reasoning about individuals is simpler than reasoning about coalitions if and only if NP $\neq$ PSPACE. For perfect information models, the complexity of the satisfiability problem for the individual fragment is not simpler, it remains PSPACE-complete.

11 ALTERNATING-TIME TEMPORAL LOGIC

Coalition Logic allows one to express strategic properties of multi-agent systems, where these systems are essentially modeled as extensive games of almost perfect information. The basic modal expression $[C]\varphi$ states that coalition C has a joint strategy for ensuring φ in the next state. What is lacking are more expressive temporal operators which allow us to describe, e.g., that coalition C has a strategy for achieving φ some time in the future. In other words, we are looking for the strategic coalitional analogue of a rich temporal logic like CTL (see Chapter 11 and [19]). Alternating-time temporal logic (ATL) [3] is precisely this temporal extension of Coalition Logic.

As usual, we start by presenting the syntax and semantics of ATL in Section 11.1. After discussing a modelling example in Subsection 11.2, we discuss axiomatisation and complexity (Subsection 11.3) and end with some extensions of ATL in Subsection 11.4.

11.1 Syntax and Semantics

Given a set of atomic propositions Π, an ATL formula φ can have the following syntactic form:

$$\varphi := \quad \bot \mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \langle\!\langle C \rangle\!\rangle \bigcirc \varphi \mid \langle\!\langle C \rangle\!\rangle \Box \varphi \mid \langle\!\langle C \rangle\!\rangle \varphi_1 \,\mathcal{U}\, \varphi_2$$

where $p \in \Pi$ and $C \subseteq N = \{1, \ldots, k\}$, the set of all agents. We define $\top$, $\wedge$, $\rightarrow$ and $\leftrightarrow$ as usual. The formula $\langle\!\langle C \rangle\!\rangle \bigcirc \varphi$ expresses that coalition C has a joint strategy for achieving φ at the next state. Thus, $\langle\!\langle C \rangle\!\rangle \bigcirc \varphi$ corresponds to $[C]\varphi$ in Coalition Logic. $\langle\!\langle C \rangle\!\rangle \Box \varphi$ expresses that coalition C can cooperate to maintain φ forever (always in the future), and $\langle\!\langle C \rangle\!\rangle \varphi_1 \,\mathcal{U}\, \varphi_2$ expresses that C can maintain φ_1 until φ_2 holds. In the standard way, we use $\Diamond\varphi$ to abbreviate $\top \,\mathcal{U}\, \varphi$.

In the same way in which CTL can be extended to CTL*, the language of ATL can be generalised to ATL*. We simultaneously define the set of ATL* *state formulas* φ and the set of ATL* *path formulas* ψ as follows:

$$\begin{array}{rl} \varphi := & \bot \mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \langle\!\langle C \rangle\!\rangle \psi \\ \psi := & \varphi \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \bigcirc \psi \mid \psi_1 \,\mathcal{U}\, \psi_2 \end{array}$$

where $p \in \Pi$ and $C \subseteq N$. Note that in ATL*, $\langle\!\langle C \rangle\!\rangle \Box \psi$ is expressible as $\langle\!\langle C \rangle\!\rangle \neg(\top \,\mathcal{U}\, \neg\psi)$ which is not an ATL formula. In general, ATL is a proper fragment of ATL*, containing only formulas where every temporal operator is immediately preceded by a cooperation modality. For $|N| = 1$, ATL=CTL and ATL*=CTL*. Given that CTL is less expressive than CTL*, it also follows that ATL is less expressive than ATL*.

The semantics of Alternating-time Temporal Logic uses concurrent game structures, essentially the coalition models we discussed for Coalition Logic. A *concurrent game structure* is a tuple $\mathcal{S} = (k, Q, \Pi, \pi, d, \delta)$, where k is the number of players ($N = \{1, \dots, k\}$),[2] Q is the set of states (usually assumed to be finite), Π is the set of atomic propositions, and $\pi : Q \to \mathcal{P}(\Pi)$ is the valuation function. For every agent $i \in N$ and every state $q \in Q$, $d_i(q) \geq 1$ gives the number of actions available to player i at state q. Hence, at state q, a *move vector* $(j_1, \dots, j_k) \in D(q) = \{1, \dots, d_1(q)\} \times \dots \times \{1, \dots, d_k(q)\}$ corresponds to a joint action at state q. Finally, for each such move vector, $\delta(q, j_1, \dots, j_k) \in Q$ is the transition function.

Like coalition models in Coalition Logic, different types of concurrent game structures correspond to natural classes of games. *Turn-based synchronous* game structures are extensive game forms of perfect information where only a single player can choose at each state. In *synchronous* game structures, the state space is the Cartesian product of the players' local state spaces. *Turn-based asynchronous* game structures involve a scheduler who determines the player who can choose the next state. Furthermore, fairness constraints can be added to these structures.

For two states $q, q' \in Q$ and an agent $i \in \Sigma$, we say that state q' is a *successor* of q if there exists a move vector $(j_1, \dots, j_k) \in D(q)$ such that $\delta(q, j_1, \dots, j_k) = q'$. A *computation* of $\mathcal{S}$ is an infinite sequence of states $\lambda = q_0, q_1, \dots$ such that for all $u > 0$, the state q_u is a successor of q_{u-1}. A computation λ starting in state q is referred to as a *q-computation*; if $u \geq 0$, then we denote by $\lambda[u]$ the u'th state in λ; similarly, we denote by $\lambda[0, u]$ and $\lambda[u, \infty]$ the finite prefix $q_0, \dots, q_u$ and the infinite suffix $q_u, q_{u+1}, \dots$ of λ respectively.

A *strategy* f_i for an agent $i \in N$ is a total function f_i mapping every finite nonempty sequence of states λ to a natural number such that if the last state of λ is q, $1 \leq f_i(\lambda) \leq d_i(q)$. Given a set $C \subseteq N$ of agents, and an indexed set of strategies $F_C = \{f_i | i \in C\}$, one for each agent $i \in C$, we define $out(q, F_C)$ to be the set of possible computations that may occur if every agent $a \in C$ follows the corresponding strategy f_a, starting when the system is in state $q \in Q$. Formally, $\lambda = q_0, q_1, \dots \in out(q, F_C)$ iff $q = q_0$ and for all $m \geq 0$, there exists a move vector $(j_1, \dots, j_k) \in D(q_m)$ such that $\delta(q_m, j_1, \dots, j_k) = q_{m+1}$ and for all $i \in C$, $j_i = f_i(\lambda[0, m])$. The semantics of ATL* can now be defined as follows.

[2]given $|N| = k$, we will sometimes use the equivalent representation $\mathcal{S} = (N, Q, \Pi, \pi, d, \delta)$

For state formulas we define

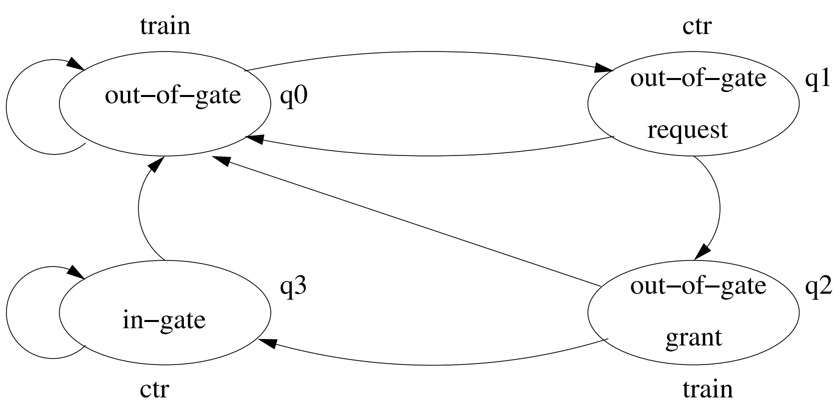
$$\begin{array}{ll}
\mathcal{S}, q \not\models \bot & \\
\mathcal{S}, q \models p & \text{iff } p \in \Pi \text{ and } p \in \pi(q) \\
\mathcal{S}, q \models \neg\varphi & \text{iff } \mathcal{S}, q \not\models \varphi \\
\mathcal{S}, q \models \varphi_1 \vee \varphi_2 & \text{iff } \mathcal{S}, q \models \varphi_1 \text{ or } \mathcal{S}, q \models \varphi_2 \\
\mathcal{S}, q \models \langle\langle C \rangle\rangle \psi & \text{iff } \exists F_C \forall \lambda \in out(q, F_C) : \mathcal{S}, \lambda \models \psi,
\end{array}$$

and for path formulas we define

$$\begin{array}{ll}
\mathcal{S}, \lambda \models \varphi & \text{iff } \mathcal{S}, \lambda[0] \models \varphi, \text{ where } \varphi \text{ is a state formula} \\
\mathcal{S}, \lambda \models \neg\psi & \text{iff } \mathcal{S}, \lambda \not\models \psi \\
\mathcal{S}, \lambda \models \psi_1 \vee \psi_2 & \text{iff } \mathcal{S}, \lambda \models \psi_1 \text{ or } \mathcal{S}, \lambda \models \psi_2 \\
\mathcal{S}, \lambda \models \bigcirc\psi & \text{iff } \mathcal{S}, \lambda[1, \infty] \models \psi \\
\mathcal{S}, \lambda \models \psi_1 \mathcal{U} \psi_2 & \text{iff } \exists m \geq 0 : (\mathcal{S}, \lambda[m, \infty] \models \psi_2 \text{ and } \forall l (0 \leq l < m \Rightarrow \mathcal{S}, \lambda[l, \infty] \models \psi_1))
\end{array}$$

11.2 An Example

The following example from [3] presents a turn-based synchronous game structure modeling a simple train system involving a train and a controller.



Formally, the concurrent game structure $\mathcal{S} = (k, Q, \Pi, \pi, d, \delta)$ consists of $\Pi = \{out{-}of{-}{-}gate, in{-}gate, request, grant\}$, $N = \{train, ctr\}$ and $Q = \{q_0, q_1, q_2, q_3\}$, with valuation function as given (e.g. $\pi(q_0) = \{out{-}of{-}gate\}$). The concurrent game structure is turn-based synchronous, so if we take the train as player 1 and the controller as player 2, we have, e.g., $d_1(q_0) = 2$ and $d_2(q_0) = 1$. The transition function at state q_0 is then described by $\delta(q_0, 1, 1) = q_0$ and $\delta(q_0, 2, 1) = q_1$.

ATL can be utilised to describe properties of this model. For instance, the formula $\langle\langle\emptyset\rangle\rangle \Box(in{-}gate \rightarrow \langle\langle ctr \rangle\rangle \bigcirc out{-}of{-}gate)$ expresses that whenever the train is in the gate, the controller can force it out immediately. Similarly, $\langle\langle\emptyset\rangle\rangle \Box(out{-}of{-}gate \rightarrow \langle\langle ctr, train \rangle\rangle \Diamond in{-}gate)$ states that whenever the train is out of the gate, it can cooperate with the controller to enter eventually. As a final example,

$$\langle\langle\emptyset\rangle\rangle \Box(out{-}of{-}gate \rightarrow \langle\langle train \rangle\rangle \Diamond(request \wedge \langle\langle ctr \rangle\rangle \Diamond grant \wedge \langle\langle ctr \rangle\rangle \Box \neg grant))$$

expresses that whenever the train is out of the gate, it can eventually send a request to enter, and the controller can then either grant it eventually or not. All these formulas are valid in the given game structure.

11.3 Axiomatics and Complexity

The axiomatisation of ATL given below is an extension of the axiomatisation given for Coalition Logic. The next-time $\bigcirc$ operator is characterised by the axioms of Coalition Logic, and the long-term temporal operators $\Box$ and $\mathcal{U}$ are captured using two fixpoint axioms each.

Formally, **ATL** is the smallest set of formulas which contains all propositional tautologies together with all instances of the axiom schemas listed in Figure 22, and which is closed under the inference rules of Modus Ponens, Monotonicity and Necessitation given below:

$$\frac{\varphi \quad \varphi \to \psi}{\psi} \qquad\qquad \frac{\varphi \to \psi}{\langle\langle C\rangle\rangle \bigcirc \varphi \to \langle\langle C\rangle\rangle \bigcirc \psi} \qquad\qquad \frac{\varphi}{\langle\langle \emptyset\rangle\rangle \Box \varphi}$$

For the case of $\Box$, $FP\Box$ states that $\langle\langle C\rangle\rangle \Box\varphi$ is a fixpoint of the operator $F(X) = \varphi \wedge \langle\langle C\rangle\rangle \bigcirc X$, and $GFP\Box$ states that it is the greatest fixpoint of $F(X)$. Analogously for the least fixpoint with $\mathcal{U}$.

$(\bot)$	$\neg\langle\langle C\rangle\rangle \bigcirc \bot$
$(\top)$	$\langle\langle C\rangle\rangle \bigcirc \top$
$(\mathtt{N})$	$(\neg\langle\langle \emptyset\rangle\rangle \bigcirc \neg\varphi \to \langle\langle N\rangle\rangle \bigcirc \varphi)$
$(\mathtt{S})$	$(\langle\langle C_1\rangle\rangle \bigcirc \varphi_1 \wedge \langle\langle C_2\rangle\rangle \bigcirc \varphi_2) \to \langle\langle C_1 \cup C_2\rangle\rangle \bigcirc (\varphi_1 \wedge \varphi_2)$ where $C_1 \cap C_2 = \emptyset$
$(FP\Box)$	$\langle\langle C\rangle\rangle \Box\varphi \leftrightarrow \varphi \wedge \langle\langle C\rangle\rangle \bigcirc \langle\langle C\rangle\rangle \Box\varphi$
$(GFP\Box)$	$\langle\langle \emptyset\rangle\rangle \Box(\psi \to (\varphi \wedge \langle\langle C\rangle\rangle \bigcirc \psi)) \to \langle\langle \emptyset\rangle\rangle \Box(\psi \to \langle\langle C\rangle\rangle \Box\varphi)$
$(FP\mathcal{U})$	$\langle\langle C\rangle\rangle \varphi_1 \mathcal{U} \varphi_2 \leftrightarrow \varphi_2 \vee (\varphi_1 \wedge \langle\langle C\rangle\rangle \bigcirc \langle\langle C\rangle\rangle \varphi_1 \mathcal{U} \varphi_2)$
$(LFP\mathcal{U})$	$\langle\langle \emptyset\rangle\rangle \Box((\varphi_2 \vee (\varphi_1 \wedge \langle\langle C\rangle\rangle \bigcirc \psi)) \to \psi) \to \langle\langle \emptyset\rangle\rangle \Box(\langle\langle C\rangle\rangle \varphi_1 \mathcal{U} \varphi_2 \to \psi)$

Figure 22. The axiom schemas of ATL

THEOREM 49 ([26]). **ATL** *is sound and complete with respect to the class of all concurrent game structures.*

The complexity of model checking ATL formulas has been investigated in [3]. As with Game Logic, given a formula φ and a finite model $\mathcal{S}$, we are interested in the complexity of determining the states of $\mathcal{S}$ where φ holds. The results are for general concurrent game structures.

THEOREM 50 ([3]). *Given an ATL formula φ and a concurrent game structure $\mathcal{S}$ with m transitions, model checking can be done in time $O(m \times |\varphi|)$. For ATL* formulas, model checking is 2EXPTIME-complete.*

The complexity of the satisfiability problem has been investigated only more recently. At the time of writing, only the complexity of ATL has been determined. Let us say that an ATL-formula φ is over a set of agents N if all coalitions mentioned in φ are subsets of N. Moreover, we say that a concurrent game structure $\mathcal{S} = (M, Q, \Pi, \pi, d, \delta)$ is over N if $M = N$ (see Footnote 2)

THEOREM 51 ([103, 26]). *Let N be a finite set of players. Then, the complexity of the satisfiability problem for ATL-formulas over N with respect to concurrent game structures over N is EXPTIME-complete.*

To demonstrate that Theorem 51 marks ongoing work in a lively research area, note that the decision procedure of [103, 26] is 2EXPTIME if the set of agents N in not fixed in advance. This gives rise to the following two related questions:

1. Is the following problem in EXPTIME: Given an ATL-formula φ, is φ satisfiable in an ATL-model over a set of agents containing at least the agents occurring in φ?

2. Is the following problem in EXPTIME: Given a set of agents N and an ATL-formula φ over N, is φ satisfiable in an ATL-model over N?

Positive answers to both questions are given in [106].

11.4 Extensions: μ-calculus and imperfect information

CTL and CTL* are both subsumed by the very expressive μ-calculus. Similarly for ATL and ATL*, one can develop an analogous *alternating μ-calculus* with general fixpoint expressions. Formally, we have a set of propositional variables V, and formulas

$$\varphi := \quad \bot \mid p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid X \mid \langle\!\langle C \rangle\!\rangle \bigcirc \varphi \mid \mu X.\varphi$$

where $X \in V$ and in $\mu X.\varphi$, X occurs free only under an even number of negations in φ. The semantics of the fixpoint operator is defined as $(\mu X.\varphi)^{\mathcal{S}} = \bigcap\{Q_0 \subseteq Q | \varphi[X := Q_0]^{\mathcal{S}} \subseteq Q_0\}$. Corresponding to the least fixpoint μX there is also a greatest fixpoint νX defined as $\nu X.\varphi = \neg\mu X.\neg\varphi$. The alternating μ-calculus subsumes both ATL and ATL*. For ATL, note that $\langle\!\langle C \rangle\!\rangle \Box\varphi = \nu X.\varphi \wedge \langle\!\langle C \rangle\!\rangle \bigcirc X$ and $\langle\!\langle C \rangle\!\rangle \varphi \,\mathcal{U}\, \psi = \mu X.\psi \vee (\varphi \wedge \langle\!\langle C \rangle\!\rangle \bigcirc X)$. In fact, it can be shown that the alternating μ-calculus is strictly more expressive than ATL*. As in the standard modal μ-calculus and in Game Logic, the complexity of model checking depends on the alternation depth of a formula, i.e., the nesting depth of alternating least and greatest fixpoints.

Various attempts have been made to introduce imperfect information into concurrent game structures. Since at the time of writing this problem is still under discussion, we restrict ourselves here to a rough sketch of some of the difficulties. In [3], a set of observable propositions is associated with every player. This extension of ATL introduces quite a few new complexities, syntactically as well as semantically, and this is witnessed by the result that model-checking becomes undecidable. Syntactically, not all ATL expressions make sense anymore, since, e.g., $\langle\!\langle C \rangle\!\rangle \Box p$ presupposes that the cooperative goal p is actually observable by the members of C. Semantically, a player's strategy has to be restricted in such a way that it can only influence propositions observable by that player. The difficulties which arise when trying to extend ATL to incomplete information also become visible in an alternative approach explored in [92]. Here, concurrent game structures are augmented with an epistemic accessibility relation $\sim_i$ for each player i. In contrast to the previous approach, the language of ATL is also extended with an epistemic K_i knowledge operator with its standard definition. The resulting *Alternating-time Temporal Epistemic Logic* (ATEL) can express properties from a variety of domains, e.g. confidentiality properties like $\langle\!\langle \{1,2\} \rangle\!\rangle(\neg K_3 p \,\mathcal{U}\, K_2 p)$. The semantics of ATL, however,

cannot simply be left unchanged. At a state q, player 1 may have some strategy s_1 to eventually achieve φ. But if he cannot distinguish q from q' where only a different strategy s_2 achieves φ, player 1 does *not* have a strategy at q (as defined for imperfect information games) for achieving φ. More precisely, player 1 has a strategy only *de dicto*, but not *de re*, and it is the de re strategies which game theory is interested in [39]. A further approached is presented in [72].

12 CONCLUSION

What can modal logic contribute to the study of strategic interaction? Four answers, not mutually exclusive, suggest themselves to us. First, logic contributes a more abstract and hence more general perspective on games. As we have seen in Sections 3 and 6, games can essentially be viewed as special kinds of Kripke models, with move relations and possibly epistemic uncertainty relations to model players' knowledge. In this way, certain game-theoretic notions such as perfect recall turn out to be special cases of general logical axioms (cf. 6.1) which have been investigated in more generality in modal logic. Similarly, game logic (Section 9) presents games as generalisations of programs, providing a semantics which is general enough to study the difference between 2-player games and 1-player games (i.e., programs). This more general perspective provided by logic also raises new questions. In the case of game logic, for instance, we may ask what operations on games suffice to build all games in a particular class of games. More generally, to the best of our knowledge, operations on games are rarely investigated in game theory, while a computational logic perspective naturally suggests such an investigation.

Second, there is the analysis of players' knowledge and beliefs in games. While much of the game-theoretic analysis of interactive epistemics has been independent of developments in epistemic logic (Section 4), the models employed are essentially the same (information partitions or Kripke models satisfying the S5 axioms), as is the notion of common knowledge. Developments in update logics (Section 7) and belief revision (see Chapter 21 of this handbook) have shifted the logical focus from an analysis of static epistemic situations to epistemic dynamics, without a doubt important in the analysis of games of imperfect information. Furthermore, the epistemic foundation of solution concepts (Section 8) translates solution concepts into the language of the epistemic logician, while the language itself remains hidden from view. More precisely, while epistemic logic rests on the link between the semantic model of knowledge and the formal language used to describe it, game theory has mainly focused on the semantic model exclusively. Only more recently have syntactic approaches to the epistemic foundations of solution concepts been advanced.

The issue of syntax or formal language is, in fact, a third way in which logic can raise new issues in strategic interaction. While many game theorists may find the logician's insistence on syntax cumbersome and unnecessary, it is precisely the interplay between syntax and semantics that the logician is interested in. In general, decision makers use a certain language and conceptual apparatus to reason about the situation at hand. As a consequence, one would suspect that game theoretic models and solution concepts would be language dependent (see also [71]). More specifically, we may be interested whether a particular logical language is rich enough to define a particular solution concept such as the Nash equilibrium (see, e.g., Theorem 15 in Section 3). As mentioned above, we may wonder whether a certain set of game operations suffices to construct all games of a

particular class. Or we may be interested in the computational complexity of reasoning about information update in games. For all these questions, a syntactic approach may help.

Fourth, due to its formal language, logic becomes important when it comes to the specification and verification of multi-agent systems. Logics like Alternating-time Temporal Logic (Section 11) have been devised as a way to specify properties of systems of interacting agents. By analysing the complexity of model checking, we find out how complex certain game-theoretic properties are to verify for a given game or game form. Hence, logic is also useful when applying game theory to complex games played by artificial agents, and hence modal logic can serve as a tool of computation for difficult real or artificial life games.

While we hope to have given some insights into how modal logic may enrich game theory, we should point out again that there are many cases where game theory can be useful for modal logic. Maybe the most interesting example is Independence-Friendly Modal Logic (IFML) [78]. In IFML, we consider formulas like $\Box\Diamond^* p$, where the modal diamond $\Diamond^*$ is independent of the box. Such a formula will be true just in case the diamond successor satisfying p can be chosen independently from the earlier box successor, in other words, there needs to be a uniform diamond successor satisfying p for all earlier box successors. As this example suggests, the semantics of IFML can be formulated in terms of imperfect information games. IFML is interesting from a logical point of view since it is more expressive than standard modal logic, due to its ability to express certain weak confluence properties.

The area of modal logic and games is active and in full development. Due to upcoming technologies like on-line auctions and e-voting, researchers apply a new range of tools and techniques to ask and settle 'standard' questions of logicians and computer scientists regarding the specification, verification and synthesis of interactive systems or mechanisms. In the 'classical' areas of game theory and social choice theory, this logical work generates both interesting results regarding formalisation and computational complexity, and at the same time new and sometimes even philosophical questions about the nature of games, or more generally, interaction.

ACKNOWLEDGEMENTS

We are thankful to Johan van Benthem, Giacomo Bonanno, Govert van Drimmelen, Valentin Goranko and Michael Wooldridge, who gave valuable suggestions on earlier drafts of this chapter.

BIBLIOGRAPHY

[1] J. Abdou and H. Keiding. *Effectivity Functions in Social Choice*. Kluwer, 1991.

[2] C.E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

[3] R. Alur, T. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49:672–713, 2002.

[4] R.J. Aumann. Game theory. In J. Eatwell, M. Milgate, and P. Newman, editors, *Game Theory*, The New Palgrave, pages 1–54. Macmillan, 1997.

[5] R.J. Aumann. Interactive epistemology I: Knowledge. *International Journal of Game Theory*, 28:263–300, 1999.

[6] R.J. Aumann and A. Brandenburger. Epistemic conditions for Nash equilibrium. *Econometrica*, (63):1161–1180, 1995.

[7] A. Baltag and L. Moss. Logics for epistemic programs. *Synthese*, 139(2):165–224, 2004. In the section 'Knowledge, Rationality and Action'.

[8] A. Baltag, L. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. In I. Gilboa, editor, *Proceedings of TARK*, pages 43–56, 1998.

[9] P. Battigalli and G. Bonanno. Recent results on belief, knowledge and the epistemic foundations of game theory. *Research in Economics*, 53:149–225, 1999.

[10] D. Berwanger and G. Lenzi. The variable hierarchy of the μ-calculus is strict. In *STACS 2005*, pages 97–109, 2005.

[11] K. Binmore. *Fun and Games – A Text on Game Theory*. W.C. Heath Co., 1992.

[12] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[13] G. Bonanno. Modal logic and game theory: Two alternative approaches. *Risk Decision and Policy*, 7(3):309–324, 2002.

[14] G. Bonanno. A characterisation of von neumann games in terms of memory. *Synthese*, pages 281–295, 2004. In the section 'Knowledge, Rationality and Action'.

[15] G. Bonanno. A simple modal logic for belief revision. *Synthese*, pages 193–228, 2005. In the section 'Knowledge, Rationality and Action'.

[16] J.R. Büchi. Using determinacy to eliminate quantifiers. In M. Karpinski, editor, *Fundamentals of Computation Theory*, volume 56 of *Lecture Notes in Computational Science*, pages 367–378. Springer, 1977.

[17] B. Chellas. *Modal Logic - An Introduction*. Cambridge University Press, 1980.

[18] B. de Bruin. *Explaining Games– On the Logic of Game Theoretic Explanations*. PhD thesis, University of Amsterdam, Amsterdam, 2004. ILLC series DS-2004-03.

[19] E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 995–1072. MIT Press, 1990. Volume B.

[20] K. Engelhardt, R. van der Meyden, and Y. Moses. Knowledge and the logic of local propositions. In *Proceedings of the 1998 Conference on Theoretical Aspects of Reasoning about Knowledge (TARK98)*, pages 29–41, 1998.

[21] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning About Knowledge*. The MIT Press, Cambridge, MA, 1995.

[22] P. Gammie and R. van der Meyden. MCK: Model checking the logic of knowledge. In R. Alur and D. Peled, editors, *Proceedings of the 16th International conference on Computer Aided Verification (CAV 2004)*, pages 479–483. Springer, 2004.

[23] J.D. Gerbrandy and W. Groeneveld. Reasoning about information change. *Journal of Logic, Language, and Information*, 6:147–169, 1997.

[24] A. Gibbard. A Pareto-consistent libertarian claim. *Journal of Economic Theory*, 7(4):388–410, 1974.

[25] V. Goranko. The basic algebra of game equivalences. *Studia Logica*, 75(2):221–238, 2003.

[26] V. Goranko and G. van Drimmelen. Decidability and complete axiomatization of the alternating-time temporal logic. *Theoretical Computer Science*, 353(1-3):93–117, 2006.

[27] Raymond Greenlaw, H. James Hoover, and Walter L. Ruzzo. *Limits to Parallel Computation: P-Completeness Theory*. Oxford University Press, Oxford & New York, 1995.

[28] J. Halpern. A computer scientist looks at game theory. *Games and Economic Behavior*, 45(1):114–131, 2003.

[29] J. Halpern and M. Vardi. The complexity of reasoning about knowledge and time. In *Proceedings 18th ACM Symposium on Theory of Computing*, pages 304–315, 1986.

[30] J. Halpern and M. Vardi. The complexity of reasoning about knowledge and time, I: Lower bounds. *Journal of Computer and System Sciences*, 38(1):195–237, 1989.

[31] J.Y. Halpern, R. van der Meyden, and M.Y. Vardi. Complete axiomatizations for reasoning about knowledge and time. *SIAM Journal on Computing*, 33(3):674–703, 2004.

[32] H. Hansen and M. Pauly. Axiomatising Nash-consistent coalition logic. In S. Flesca, S. Greco, N. Leone, and G. Ianni, editors, *Logics in Artificial Intelligence*, LNAI 2424, pages 394–406. Springer, 2002.

[33] D. Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, pages 497–604. Kluwer, 1984.

[34] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. The MIT Press: Cambridge, MA, 2000.

[35] B.P. Harrenstein, W. van der Hoek, J.-J.Ch. Meyer, and C. Witteveen. A modal characterization of Nash equilibrium. *Fundamenta Informaticae*, 57(2–4):281–321, 2003.

[36] J. Hintikka. *Knowledge and Belief*. Cornell University Press, 1962. Republished King's College London Publications, 2005.

[37] J. Hintikka. *Logic, Language Games and Information*. Oxford University Press, 1973.

[38] W. van der Hoek and M. Wooldridge. Model checking knowledge and time. In D. Bošnački and S. Leue, editors, *Model Checking Software, Proceedings of SPIN 2002 (LNCS Volume 2318)*, pages 95–111. Springer, 2002.

[39] W. Jamroga and W. van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63(2-3):185–219, 2004.

[40] E. Kindler. Safety and liveness properties: A survey. *Bulletin of the European Association for Theoretical Computer Science*, 53:268–272, 1994.

[41] D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[42] H.W. Kuhn. Extensive games and the problem of information. In H.W. Kuhn and A.W. Tucker, editors, *Contributions to the Theory of Games, Volume II*, Annals of Mathematics Studies 28, pages 193–216. Princeton University Press, 1953.

[43] L. Lamport. Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering*, SE-3(2), 1977.

[44] M. Lange and C. Stirling. Model checking games for branching time logics. *Journal of Logic and Computation*, 12(4):623–639, 2002.

[45] Logic and the foundations of game and decision theory (LOFT). `http://www.econ.ucdavis.edu/faculty/bonanno/loft.html`.

[46] A. Lomuscio and F. Raimondi. MCMAS: A model checker for multi-agent systems. In *Proceedings of TACAS06*, volume 3920, pages 450–454. Springer Verlag, 2006.

[47] A. Lomuscio, R. van der Meyden, and M. Ryan. Knowledge in multiagent systems: initial configurations and broadcast. *ACM Transactions on Computational Logic (TOCL)*, 1(2):247–284, 2000.

[48] A.R. Lomuscio. *Knowledge Sharing among Ideal Agents*. PhD thesis, University of Birmingham, Birmingham, UK, 1999.

[49] J.-J.Ch. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science*. Cambridge University Press, Cambridge, England, 1995.

[50] M. Mizutani, Y. Hiraide, and H. Nishino. Computational complexity to verify the instability of effectivity function. *International Journal of Game Theory*, 22:225–239, 1993.

[51] R.C. Moore. A formal theory of knowledge and action. In J.F. Allen J., Hendler, and A. Tate, editors, *Readings in Planning*, pages 480–519. Morgan Kaufmann Publishers, San Mateo, CA, 1990.

[52] H. Moulin. *The Strategy of Social Choice*. North-Holland, 1983.

[53] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, 1951.

[54] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. John Wiley and Sons, 1944.

[55] D. Niwiński. μ-calculus via games. In J.C. Bradfield, editor, *Computer Science Logic*, volume 2471 of *Lecture Notes in Computer Science*, pages 27–43. Springer, 2002.

[56] M. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[57] C. Papadimitriou. Algorithms, games, and the Internet. In *33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 749–753, 2001.

[58] R. Parikh. Propositional logics of programs: New directions. In M. Karpinski, editor, *Foundations of Computation Theory*, LNCS 158, pages 347–359. Springer, 1983.

[59] R. Parikh. The logic of games and its applications. In M. Karpinski and J. van Leeuwen, editors, *Topics in the Theory of Computation*, Annals of Discrete Mathematics 24, pages 111–140. Elsevier, 1985.

[60] R. Parikh and R. Ramanujam. *Distributed Processes and the Logic of Knowledge*, pages 256–268. Number 193 in Lecture Notes in Computer Science. Springer, 1985.

[61] M. Pauly. From programs to games: Invariance and safety for bisimulation. In P. Clote and H. Schwichtenberg, editors, *Computer Science Logic*, LNCS 1862. Springer, 2000.

[62] M. Pauly. *Logic for Social Software*. PhD thesis, University of Amsterdam, 2001.

[63] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[64] M. Pauly. On the complexity of coalitional reasoning. *International Game Theory Review*, 4(3):237–254, 2002.

[65] M. Pauly and R. Parikh. Game logic - an overview. *Studia Logica*, 75(2):165–182, 2003.

[66] B. Peleg. Effectivity functions, game forms, games and rights. *Social Choice and Welfare*, 15:67–80, 1998.

[67] M. Piccione and A. Rubinstein. On the interpretation of decision problems with imperfect recall. *Games and Economic Behaviour*, 20(1):3–24, 1997.

[68] J.A. Plaza. Logics of public communications. In M.L. Emrich, M.S. Pfeifer, M. Hadzikadic, and Z.W. Ras, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216, 1989.

[69] W.V.O. Quine. Quantifiers and propositional attitudes. In L. Linsky, editor, *Reference and Modality*, pages 101–111. Oxford University Press, London, 1971.

[70] P. Reny. *Two Papers on the Theory of Strategic Behaviour.* PhD thesis, Princeton University, Princeton, USA, 1988.

[71] A. Rubinstein. *Economics and Language.* Cambridge University Press, 2000.

[72] P.Y. Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2), 2004.

[73] W. Spohn. How to make sense of game theory. In W. Balzer, W. Spohn, and W. Stegmüller, editors, *Studies in Contemporary Economics, (Vol 2: Philosophy of Economics)*, pages 239–270. Springer, Berlin, 1982.

[74] R. Stalnaker. On the evaluation of solution concepts. *Theory and Decisions*, 37:49–73, 1994.

[75] R. Stalnaker. Belief and counterfactual reasoning in games. *Economics and Philosophy*, 12:133–163, 1996.

[76] R. Stalnaker. Belief revision in games: Forward and backward induction. *Mathematical Social Sciences*, 36:31–56, 1998.

[77] Theoretical aspects of reasoning about knowledge (TARK). `http://www.tark.org`.

[78] T. Tulenheimo. On IF modal logic and its expressive power. In *Advances in Modal Logic, Volume 4*, P. Balbiani *et al.*, eds, pp. 475–498. King's College Publications, 2003.

[79] K. Tuyls, A. Nowe, T. Lenaerts, and B. Manderick. An evolutionary game theoretic perspective on learning in multi-agent systems. *Synthese*, pages 297–330, 2004. In the section 'Knowledge, Rationality and Action'.

[80] J. van Benthem. *Modal Correspondence Theory.* PhD thesis, University of Amsterdam, 1976.

[81] J. van Benthem. *Exploring Logical Dynamics.* CSLI Publications, Stanford, USA, 1996.

[82] J. van Benthem. Update as relativization. Technical report, Institute for Logic, Language and Computation, Amsterdam, 1999. `http://staff.science.uva.nl/~johan/Upd-Rel.pdf`.

[83] J. van Benthem. Games in dynamic-epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2001.

[84] J. van Benthem. Logics for information update. In J. van Benthem, editor, *Proceedings of TARK VIII*, pages 51–88, Los Altos, 2001. Morgan Kaufmann.

[85] J. van Benthem. Extensive games as process models. *Journal of Logic, Language and Information*, 11(3):289–313, 2002.

[86] J. van Benthem. One is a lonely number: the logic of communication. In P. Koepke and W. Pohlers, editors, *Logic Colloquium '02*, pages 96–129. Assocation for Symbolic Logic, 2002.

[87] J. van Benthem. Logic games are complete for game logics. *Studia Logica*, 75(2):183–203, 2003.

[88] J. van Benthem. Rational dynamics and epistemic logic in games. In S. Vannucci, editor, *Logic, Games and Social Choice*, Siena, 2003. Siena University Press. Extended version to appear in *International Journal of Game Theory.*

[89] J. van Benthem and E. Pacuit. The tree of knowledge in action: Toward a common perspective. Technical report, ILLC, the University of Amsterdam, 2006.

[90] J. van Benthem, J. van Eijck, and B.P. Kooi. Common knowledge in update logics. In R. van der Meyden, editor, *Theoretical aspects of rationality and knowledge: proceeding of the tenth conference (TARK 2005)*, pages 253–261, 2005.

[91] W. van der Hoek, J. Jaspars, and E. Thijsse. Theories of knowledge and ignorance. In S. Rahman, J. Symons, D.M. Gabbay, and J.P. van Bendegem, editors, *Logic, Epistemology and the Unity of Scienc*, pages 381–418. Kluwer, 2004.

[92] W. van der Hoek and M Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.

[93] R. van der Meyden. Axioms for knowledge and time in distributed systems with perfect recall. In *Proceedings of the Ninth Annual IEEE Symposium on Logic in Computer Science (LICS-94)*, pages 448–457, Paris, France, July 1994.

[94] R. van der Meyden and N. Shilov. Model checking knowledge and time in systems with perfect recall. In *Proceedings of the Conference on Foundations of Software Technology and Theoretical Computer Science (LNCS Volume 1738)*, pages 432–445. Springer, 1999.

[95] R. van der Meyden and K. Su. Symbolic model checking the knowledge of the dining cryptographers. In *17th IEEE Computer Security Foundations Workshop*, pages 280–291, 2004.

[96] R. van der Meyden and M. Vardi. Synthesis form knowledge-based specifications. In *Proceedings CONCUR'98*, number 1466, pages 34–49, 1998.

[97] R. van der Meyden and K. Wong. Complete axiomatisations for reasoning about knowledge and branching time. *Studia Logica*, 75(1):93–123, 2003.

[98] H.P. van Ditmarsch. *Knowledge Games.* PhD thesis, University of Groningen, Groningen, 2000.

[99] H.P. van Ditmarsch and B.P. Kooi. The secret of my success. Accepted for Synthese, 2005.

[100] H.P. van Ditmarsch, J. Ruan, and L.C. Verbrugge. Model checking sum and product. In S. Zhang and R. Jarvis, editors, *Proceedings of the 18th Australian Joint Conference on Artificial Intelligence (AI 2005)*, volume 3809 of *LNAI*, pages 790–795, 2005.

[101] H.P. van Ditmarsch, W. van der Hoek, and B.P. Kooi. Dynamic epistemic logic. manuscript, to appear in 2006.

[102] H.P. van Ditmarsch, W. van der Hoek, and B.P. Kooi. Concurrent dynamic epistemic logic. In K.F. Jorgensen V.F. Hendricks and S.A. Pederson, editors, *Knowledge Contributors*, pages 105–143. Kluwer Academic Pusblishers, 2003.

[103] G. van Drimmelen. Satisfiability in alternating-time temporal logic. In *Eighteenth Annual IEEE Symposium on Logic in Computer Science (LICS 2003)*, pages 208–217, Ottawa, Canada, 2003.

[104] J. van Eijck. DEMO: a system for dynamic epistemic modeling. `http://homepages.cwi.nl/~jve/demo`.

[105] Y. Venema. Representing game algebras. *Studia Logica*, 75(2):239–256, 2003.

[106] D. Walther, C. Lutz, F. Wolter, and M. Wooldridge. ATL satisfiability is indeed ExpTime-complete. submitted, 2005.

[107] E. Zermelo. Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. In E.W. Hobson and A.E.H Love, editors, *Proceedings of the Fifth International Congress of Mathematicians*, pages 501–4, Cambridge, 1913. Cambridge U.P.

Handbook of Modal Logic – P. Blackburn et al. (Editors)
© 2007 Elsevier B.V. All rights reserved.

21

MODAL LOGIC AND PHILOSOPHY

Sten Lindström and Krister Segerberg

1 Alethic modal logic . . . 1150
1.1 The search for the intended interpretation . . . 1151
1.2 Carnap's formal semantics for quantified modal logic . . . 1152
1.3 Quine's interpretational challenge . . . 1156
1.4 The advent of possible worlds semantics . . . 1159
1.5 General intensional logic . . . 1175
1.6 Logical and metaphysical necessity . . . 1181
2 The modal logic of belief change . . . 1183
2.1 Introduction . . . 1184
2.2 Conditional logic . . . 1189
2.3 Update and the logic of conditionals . . . 1192
2.4 Revision and basic DDL . . . 1193
2.5 Revision and full or unlimited DDL . . . 1195
3 Logic of action and deontic logic . . . 1197
3.1 Logic of action . . . 1198
3.2 Deontic logic . . . 1204

Modal logic is one of philosophy's many children. As a mature adult it has moved out of the parental home and is nowadays straying far from its parent. But the ties are still there: philosophy is important for modal logic, modal logic is important for philosophy. Or, at least, this is a thesis we try to defend in this chapter. Limitations of space have ruled out any attempt at writing a survey of all the work going on in our field — a book would be needed for that. Instead, we have tried to select material that is of interest in its own right or exemplifies noteworthy features in interesting ways. Here are some themes which have guided us throughout the writing:

- *The back-and-forth between philosophy and modal logic.* There has been a good deal of give-and-take in the past. Carnap tried to use his modal logic to throw light on old philosophical questions, thereby inspiring others to continue his work and still others to criticise it. He certainly provoked Quine, who in his turn provided — and continues to provide — a healthy challenge to modal logicians. And Kripke's and David Lewis's philosophies are connected, in interesting ways, with their modal logic. Analytic philosophy would have been a lot different without modal logic!

- *The interpretation problem.* The problem of providing a certain modal logic with an intuitive interpretation should not be conflated with the problem of providing a formal system with a model-theoretic semantics. An intuitively appealing model-theoretic semantics may be an important step towards solving the interpretation problem, but only a step. One may compare this situation with that in probability theory, where definitions of concepts like 'outcome space' and 'random variable' are orthogonal to questions about "interpretations" of the concept of probability.

- *The value of formalisation.* Modal logic sets standards of precision, which are a challenge to — and sometimes a model for — philosophy. Classical philosophical questions can be sharpened and seen from a new perspective when formulated in a framework of modal logic. On the other hand, representing old questions in a formal garb has its dangers, such as simplification and distortion.

- *Why modal logic rather than classical (first or higher order) logic?* The idioms of modal logic — today there are many! — seem better to correspond to human ways of thinking than ordinary extensional logic. (Cf. Chomsky's conjecture that the NP + VP pattern is wired into the human brain.)

In his *An Essay in Modal Logic* [107] von Wright distinguished between four kinds of modalities: *alethic* (modes of truth: necessity, possibility and impossibility), *epistemic* (modes of being known: known to be true, known to be false, undecided), *deontic* (modes of obligation: obligatory, permitted, forbidden) and *existential* (modes of existence: universality, existence, emptiness). The existential modalities are not usually counted as modalities, but the other three categories are exemplified in three sections into which this chapter is divided. Section 1 is devoted to alethic modal logic and reviews some main themes at the heart of philosophical modal logic. Sections 2 and 3 deal with topics in epistemic logic and deontic logic, respectively, and are meant to illustrate two different uses that modal logic or indeed any logic can have: it may be applied to already existing (non-logical) theory, or it can be used to develop new theory.

1 ALETHIC MODAL LOGIC

In this part we consider the challenge that Quine posed in 1947 to the advocates of modal logic to provide an account of modal notions that is intuitively clear, allows "quantifying in", and does not presuppose intensional entities. The modal notions that Quine and his contemporaries were primarily concerned with in the 1940's were, broadly speaking, the logical modalities rather than the metaphysical ones that have since come to prevail. In the 1950's modal logicians responded to Quine's challenge by providing quantified modal logic with model-theoretic semantics of various types. In doing so they also, explicitly or implicitly, addressed Quine's interpretation problem. Here we shall consider the approaches developed by Carnap in the late 1940's, and by Kanger, Hintikka, Montague, and Kripke in the 1950's and early 1960's, and discuss to what extent these approaches were successful in meeting Quine's doubts about the intelligibility of quantified modal logic.

It is useful to divide the reactions to Quine's challenge into two periods. During the first period modal logicians provided modal logic with formal semantics as just mentioned. In the second period philosophers — inspired by the success of possible worlds

semantics — came to take the notion of a possible world seriously as a tool for philosophical analysis. Philosophical analyses in terms of possible worlds were provided for many concepts of central philosophical importance: propositional attitudes [42, 43, 45], metaphysical necessity, identity, and naming [69, 70], "intensional entities" like propositions, properties and events [84, 61, 102, 103], counterfactual conditionals and causality [77, 78], supervenience [62]. At the same time the notion of a possible world itself came in for philosophical analysis. The problems of giving a satisfactory analysis of this notion indicates that Quine's interpretational challenge is still alive. The basic philosophical questions surrounding the notions of alethic necessity and possibility are as puzzling as ever! We end this section by discussing the relationship between the logical and metaphysical interpretation of the alethic modalities.

1.1 The search for the intended interpretation

Starting with the work of C. I. Lewis, an immense number of formal systems of modal logic have been constructed based on classical propositional or predicate logic. The originators of modern modal logic, however, were not very clear about the intuitive meaning of the symbols $\Box$ and $\Diamond$, except to say that these should stand for some kind of necessity and possibility, respectively. For instance, in *Symbolic Logic* [72], Lewis and Langford write:

> It should be noted that the words "possible", "impossible" and "necessary" are highly ambiguous in ordinary discourse. The meaning here assigned to $\Diamond p$ is a *wide* meaning of "possibility" — namely, logical conceivability or the absence of self-contradiction. (160–61)

This situation led to a search for more rigorous interpretations of modal notions. Gödel [35] suggested interpreting the necessity operator $\Box$ as standing for provability (*informal provability* or, alternatively, *formal provability* in a fixed formal system), a suggestion that subsequently led to the modern *provability interpretations* of Solovay, Boolos and others.[1]

After Tarski [105, 106] had developed rigorous notions of satisfaction, truth and logical consequence for classical extensional languages, the question arose whether the same methods could be applied to the languages of modal logic and related systems. One natural idea, that occurred to Carnap in the 1940's, was to let $\Box\varphi$ be true of precisely those formulæ φ that are *logically valid* (or logically true) according to the standard semantic definition of logical validity. This idea led him to the following semantic clause for the operator of logical necessity:

> $\Box\varphi$ is true in an interpretation $\mathcal{I}$ iff φ is true in every interpretation $\mathcal{I}'$.

This kind of approach, which we may call the *validity interpretation*, was pursued by Carnap, using so-called state descriptions, and subsequently also by Kanger [53, 54] and Montague [83], using Tarski-style model-theoretic interpretations rather than state descriptions. In Hintikka's and Kanger's early work on modal semantics other interpretations of $\Box$ were also considered, especially, epistemic ('It is known that φ') and deontic ones ('It ought to be the case that φ'). In order to study these and other non-logical modalities, the introduction by Hintikka and Kanger of *accessibility relations* between

[1] Cf. [101] and [13].

possible worlds (models, domains) was crucial. Finally, Kripke [66, 67, 68] introduced the kind of model structures that are nowadays the standard formal tool for the model-theoretic study of modal and related non-classical logics: Kripke models. Thus Kripke gave possible worlds semantics its modern and mature form.

In Carnap's, Kanger's and Montague's early theories, the space of possibilities (the "possible worlds") is represented by one comprehensive collection containing *all* state descriptions, domains, or models, respectively. Hence, every state description, domain, or model is thought of as representing a genuine possibility. Hintikka, Kripke and modern possible worlds semantics are instead working with semantic interpretations in which the space of possibilities is represented by an arbitrary non-empty set **K** of model sets (in the case of Hintikka) or "possible worlds" (Kripke). Following Hintikka's [46, 47] terminology, one may say that the early theories of Carnap, Kanger, and Montague were considering *standard interpretations* only, where one quantifies over what is, in some formal sense, *all* the possibilities. In the possible worlds approach, one also considers *non-standard* interpretations, where arbitrary non-empty sets of possibilities are considered.[2] The consideration of interpretations (model structures) that are non-standard in this sense — in combination with the use of accessibility relations between worlds in each interpretation — made it possible for Kripke [64, 67, 68] to prove completeness theorems for various systems of propositional and quantified modal logic (**T**, **B**, **S4**, etc.).

1.2 Carnap's formal semantics for quantified modal logic

The proof theoretic study of quantified modal logic was pioneered by Ruth Barcan Marcus [5, 6, 7] and Rudolf Carnap [16, 17] who were the first to formulate axiomatic systems that combined quantification theory with (**S4**- and **S5**-type) modal logic. The attempts to interpret quantified modal logic by means of formal semantic methods also began with Carnap.

Carnap's project was not only to develop a semantics (in the sense of Tarski) for intensional languages, but also to use metalinguistic notions from formal semantics to throw light on the modal ones. In 'Modalities and quantification' from 1946 he writes:

> It seems to me ... that it is not possible to construct a satisfactory system before the meaning of the modalities are sufficiently clarified. I further believe that this clarification can best be achieved by correlating each of the modal concepts with a corresponding semantical concept (for example, necessity with **L**-truth).

In [16, 17] Carnap presented a formal semantics for logical necessity based on Leibniz's old idea that a proposition is necessarily true if and only if it is true in all possible worlds. Suppose that we are considering a first-order predicate language $\mathcal{L}$ with predicate symbols and individual constants, but no function symbols. In addition to Boolean connectives, quantifiers and the identity symbol $=$ (considered as a logical symbol), the language $\mathcal{L}$ also contains the modal operator $\Box$ for logical necessity. We assume that $\mathcal{L}$ comes with a *domain of individuals* D and that there is a one-to-one correspondence between the individual constants of $\mathcal{L}$ and the individuals in D. Intuitively speaking, each individual in D has exactly one individual constant as its (canonical) name. A *state description* S for $\mathcal{L}$ is simply a set of (closed) atomic sentences of the form $P(a_1, \ldots, a_n)$, where P is

[2]For the standard/non-standard distinction, see also [23].

an n-ary predicate in $\mathcal{L}$ and $a_1, \ldots, a_n$ are individual constants in $\mathcal{L}$.[3] Carnap [17, p. 9] writes "...the state descriptions represent Leibniz's possible worlds or Wittgenstein's possible states of affairs".

In order to interpret quantification, Carnap introduced the notion of an *individual concept* (relative to $\mathcal{L}$): An individual concept is simply a function f that assigns to every state description S an individual constant $f(S)$ (representing an individual in D). Intuitively speaking, individual concepts are functions from possible worlds to individuals. According to Carnap's semantics, individual variables are assigned values *relative to* state descriptions. An *assignment* is a function g that to every state description S and every individual variable x assigns an individual constant $g(x, S)$. Intuitively, $g(x, S)$ represents the individual that is the value of x under the assignment g in the possible world represented by S. We may speak of $g(x, S)$ as the *value extension* of x in S relative to g. Analogously, the individual concept $(\lambda S)g(x, S)$ that assigns to every state description S the value extension of x in S relative to g, we call the *value intension* of x relative to g. Thus, according to Carnap's semantics a variable is assigned both a *value intension* and a *value extension* [17, p. 45]. The value extension assigned to a variable in a state description S is simply the value intension assigned to the variable applied to S.

With these notions in place, we can define what it means for a formula φ of $\mathcal{L}$ to be *true* in a state description relative to an assignment g (in symbols, $S\varphi[g]$).

For atomic formulæ of the form $P(t_1, \ldots, t_n)$, where $t_1, \ldots, t_n$ are individual terms, i.e., variables or individual constants, we have:

(1) $S \models P(t_1, \ldots, t_n)[g]$ iff $P(S(t_1, g), \ldots, S(t_n, g)) \in S$.

Here, $S(t_i,\ g)$ is the extension of the term t_i in the state description S relative to the assignment g. Thus, if t_i is an individual constant, then $S(t_i, g)$ is t_i itself; and if t_i is a variable, then $S(t_i, g) = g(t_i, S)$.

The semantic clause for the identity symbol is:

(2) $S \models (t_1 = t_2)[g]$ iff $S(t_1, g) = S(t_2, g)$.

That is, the identity statement $t_1 = t_2$ is true in a state description S relative to an assignment g if and only if the terms t_1 and t_2 have the same extension in S relative to g.

The clauses for the Boolean connectives are the usual ones. Carnap's clause for the universal quantifier is:

(3) $S \models \forall x\varphi[g]$ iff for every assignment g' such that $g =_x g', S \models \varphi[g']$,

where $g =_x g'$ means that the assignments g and g' assign the same value intensions to all the variables that are distinct from x and possibly assign different value intensions to x. Intuitively, then $\forall x\varphi(x)$ may be read: "for every assignment of an individual concept to $x, \varphi(x)$".

Finally, the semantic clause for the necessity operator is the expected one:

(4) $S \models \Box\varphi[g]$ iff, for every state description $S', S' \models \varphi[g]$.

[3] Actually Carnap's state descriptions are sets of literals (i.e., either atomic sentences or negated atomic sentences) that contain for each atomic sentence either it or its negation. However, for our purposes we may identify a state description with the set of atomic sentences that it contains.

That is, the modal formula 'it is (logically) necessary that φ' is true in a state description S (relative to an assignment g) if and only if φ is true in every state description S' (relative to g).

A formula φ is *true in a state description* S (in symbols, $S \vDash \varphi$) if it is true in S relative to every assignment. *Logical truth* (logical validity) is defined as truth in all state descriptions. We write $\vDash \varphi$ for φ being logically true.

Carnap's semantics satisfies the following principles:

(5) All truth-functional tautologies are logically true.

(6) The set of logical truths is closed under modus ponens.

(7) The standard principles of quantification theory (without identity) are valid. In particular,

(US) $\forall x\varphi(x) \to \varphi(x)$ (*Universal Specification*)
(EG) $\varphi(t/x) \to \exists x\varphi$ *(Existential Generalisation)*
(where t is substitutable for x in φ)

hold without restrictions.

It is easy to verify that $\Box$ satisfies the usual laws of the system **S5**, together with the so-called Barcan formula and its converse, and the rule of necessitation:

(K)	$\vDash \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$.	
(T)	$\vDash \Box\varphi \to \varphi$	
(S4)	$\vDash \Box\varphi \to \Box\Box\varphi$.	
(S5)	$\vDash \neg\Box\varphi \to \Box\neg\Box\varphi$	
(BF)	$\vDash \forall x\Box\varphi(x) \to \Box\forall x\varphi(x)$.	(*The Barcan formula*)
(CBF)	$\vDash \Box\forall x\varphi(x) \to \forall x\Box\varphi(x)$.	(*The Converse Barcan formula*)
(Nec)	If $\vDash \varphi$, then $\vDash \Box\varphi$.	

Notice that the Barcan formula (BF) and its converse (CBF) are schemata rather than single formulæ.

The following schemata are also valid in Carnap's semantics:

(8) $\vDash \Box\varphi$ iff $\vDash \varphi$.

(9) $\vDash \neg\Box\varphi$ iff $\nvDash \varphi$.

(10) Either $\vDash \Box\varphi$ or $\vDash \neg\Box\varphi$.

For identity, we have:

(LI) $\vDash t = t$. (*Law of Identity*)

However, the unrestricted principle of *indiscernibility of identicals* is not valid in Carnap's semantics. In other words, the following principle does not hold for all formulæ φ:

$(I =)$ $\vDash \forall x\forall y(x = y \to (\varphi(x/z) \to \varphi(y/z)))$.

Instead, we have a restricted version of $(I =)$:

$(I =_{\text{restr}})$ $\vDash \forall x\forall y(x = y \to (\varphi(x/z) \to \varphi(y/z)))$, provided that φ does not contain any occurrences of $\Box$.

For the unrestricted case, we only have:

$(I\Box =) \vDash \forall x \forall y(\Box(x = y) \to (\varphi(x/z) \to \varphi(y/z)))$.

The following principle is of course not valid according to Carnap's semantics:

$(\Box =) \quad \forall x \forall y(x = y \to \Box(x = y))$. (*Necessity of Identity*)

In the presence of the other principles, it is equivalent to the unrestricted principle of indiscernibility of identicals. Nor do we have:

$(\Box \neq) \quad \forall x \forall y(x \neq y \to \Box(x \neq y))$. (*Necessity of Non-Identity*)

In view of Church's undecidablity theorem for the predicate calculus, it is easy to prove that Carnap's quantified modal logic is not axiomatizable. For every sentence φ of predicate logic φ, either $\Box\varphi$ or $\neg\Box\varphi$ is true in every state description. So, if Carnap's logic were axiomatizable, then we could decide effectively whether φ is provable in predicate logic. But this is contrary to Church's theorem.

THEOREM 1. *The set of all logically true sentences according to Carnap's semantics is not recursively enumerable, so there is no formal axiomatic system with this set as its theorems.*

Carnap introduced the notion of a *meaning postulate* to account for analytic connections between the non-logical symbols of a predicate language. Thus, suppose that *MP* is the set of all the meaning postulates of a given language $\mathcal{L}$. *MP* is then a set of sentences in the non-modal fragment of $\mathcal{L}$. We say that a state description S is *admissible* if *MP* $\cup S$ is consistent. Then, we can interpret $\Box$ as 'analytic necessity' by modifying clause (4) above to:

(4′) $S \vDash \Box\varphi$ iff, for every admissible state description $S', S' \vDash \varphi$.

We also say that φ is *analytically true* iff φ is true in all admissible state descriptions. In the modified semantics, we have:

$S \vDash \Box\varphi$ iff φ is analytically true.
$S \vDash \neg\Box\varphi$ iff φ is not analytically true.

Carnap's semantics for the quantifiers can be understood in two ways. The most straightforward interpretation is to say that the quantifiers simply range over individual concepts. Sometimes Carnap himself characterises his interpretation of the quantifiers in this way, and this is how Quine describes it. There is, however, another more subtle interpretation according to which every individual term, including the (free) variables, has a double semantic role given by its extension and its intension, respectively. Each variable has a value extension as well as a value intension. According to this interpretation — which presumably is the one that Carnap really had in mind — it is simply wrong to ask for *the* range of the individual variables. In ordinary extensional contexts the variables can be thought of as ranging over ordinary individuals. However, in intensional contexts the intensions associated with the variables come into play. This is what explains why the principle $(\Box =)$ fails.

Carnap's interpretation of the quantifiers can still be criticised for being unintuitive. The problem is that he lacks a way of discriminating between those individual concepts that, intuitively speaking, pick out one and the same individual in all possible worlds and those that don't. Suppose that we have assigned to the variable x as its value intension the individual concept: *the number of planets.* Relative to this assignment it is true that:

(1) $x = 9 \wedge \neg\Box(x = 9)$.

However, there is no *object* that has the property of being identical with 9 but doesn't have this property necessarily. So from (1) it should not follow that:

(2) $\exists x(x = 9 \wedge \neg\Box(x = 9))$.

But of, course, on Carnap's interpretation of the quantifiers, (2) is a logical consequence of (1). Intuitively, one should be able to make the inference from (1) to (2) only if the concept assigned to x in (1) is, what might be called, a *logically rigid concept*, i.e. a concept that picks out the same individual relative to every state description.[4]

1.3 Quine's interpretational challenge

Quine's criticism of quantified modal logic comes in different strands. First, there is the simple observation that classical quantification theory with identity cannot be applied to a language in which substitutivity of identicals for singular terms fails. It seems, from the so-called Morning Star Paradox, that either universal specification (US) (and its mirror image: existential generalisation (EG)) or indiscernibility of identicals, (I=), has to be given up. This observation gives rise to the following weak, and apparently uncontroversial, Quinean claim: Classical quantification theory (with identity and individual constants) cannot be combined with non-extensional operators (i.e., operators for which substitutivity of identicals for singular terms fail) without being modified in some way. This weak claim already gives rise to the challenge of extending quantification theory in a consistent way to languages with non-extensional operators.

In addition to the weak claim, there is the much stronger claim that one sometimes can find in Quine's early works, that objectual quantification into non-extensional (so called "opaque") constructions simply does not make sense [91, 93, 94]. The argument for this claim is based on the idea that occurrences of variables inside of opaque constructions do not have purely referential occurrences, i.e., they do not serve simply to refer to their objects, and cannot therefore be bound by quantifiers outside of the opaque construction. Thus quantifying into contexts governed by non-extensional operators would be like trying to quantify into quotations. This claim is hardly credible in the face of the multitude of quantified intensional logics that have been developed since it was first made, and we take it to be refuted by the work of among others, David Kaplan [59, 61] and Kit Fine [26, 27].[5]

Then, there is Quine's claim that quantified modal logic is committed to *Aristotelian essentialism*, i.e., the view that it makes sense to say of an object, quite independently of how it is described, that it has certain of its traits necessarily, and others only contingently. Aristotelian essentialism, however, comes in stronger and weaker forms. Kripke's "metaphysical necessity" of *Naming and Necessity* represents a strong form of essentialism, while there are weaker forms according to which only logical properties that are shared by all individuals are essential. A quantified modal logic needs only be committed to this weak relatively benign form of essentialism.

---

[4]The notion of a logically rigid concept is closely related Carnap's [17, Part II] notion of an L-determinate intension. Intuitively, an L-determinate intension picks out the same extension in every state description. Thus, Carnap's notion of L-determinacy may be viewed as a precursor of Kripke's notion of rigidity.

[5]See also Burgess [14] and Neale [86] for recent evaluations of Quine's criticism of quantified modal logic.

Here we shall only consider the specific criticism that Quine directed in 1947 toward quantification into contexts of logical or analytical necessity. In his paper 'The problem of interpreting modal logic' from 1947, Quine formulates what one might call *Quine's challenge* to the advocates of quantified modal logic:

> There are logicians, myself among them, to whom the ideas of modal logic (e. g. Lewis's) are not intuitively clear until explained in non-modal terms. But so long as modal logic stops short of quantification theory, it is possible ... to provide somewhat the type of explanation required. When modal logic is extended (as by Miss Barcan) to include quantification theory, on the other hand, serious obstacles to interpretation are encountered — particularly if one cares to avoid a curiously idealistic ontology which repudiates material objects.

What Quine demands of the modal logicians is nothing less than an explanation of the notions of quantified modal logic in non-modal terms. Such an explanation should satisfy the following requirements:

(i) It should be expressed in an extensional language. Hence, it cannot use any non-extensional constructions.

(ii) The explanation should be allowed to use concepts from the 'theory of meaning' like analyticity and synonymy applied to expressions of the metalanguage. Quine is, of course, quite sceptical about the intelligibility of these notions as well. But he considers it to be progress of a kind, if modal notions could be explained in these terms.

(iii) The explanation should make sense of sentences like:

$$\exists x(x \text{ is red } \wedge \Diamond(x \text{ is round})),$$

in which a quantifier outside a modal operator binds a variable within the scope of the operator and the quantifier ranges over ordinary physical objects (in distinction from Frege's "Sinne" or Carnap's "individual concepts"). In other words, the explanation should make sense of 'quantifying in' in modal contexts.

Quine [92] — like Carnap before him — starts out from a metalinguistic interpretation of the necessity operator $\Box$ in terms of the predicate '... is analytically true'. Disregarding possible complications in connection with the interpretation of iterated modalities, we have for sentences φ of the object language:

$$\text{`}\Box\varphi\text{' is true iff } \varphi \text{ is analytically true.}$$

Now Quine argues for the thesis that it is impossible to combine analytical necessity with a standard theory of quantification (over physical objects). The argument (a variation of "the Morning Star Paradox") is based on the premises:

(1) $\Box$(Hesperus = Hesperus)

(2) Phosphorus = Hesperus

(3) $\neg\Box$(Phosphorus = Hesperus),

where 'Phosphorus' and 'Hesperus' are two proper names (individual constants) and $\Box$ is to be read 'It is analytically necessary that'. We assume that 'Phosphorus' is used by the language community as a name for a certain bright heavenly object sometimes visible in the morning and that 'Hesperus' is used for some bright heavenly object sometimes visible in the evening. Unbeknownst to the community, however, these objects are one and the same, namely, the planet Venus. 'Hesperus = Hesperus' being an instance of the Law of Identity is clearly an analytic truth. It follows that the premise (1) is true. (2) is true, as a matter of fact. 'Phosphorus = Hesperus' is obviously not an analytic truth, 'Phosphorus' and 'Hesperus' being two different names with quite distinct uses. So, (3) is true.

From (1), (2), (3) and the Law of Identity, we infer by sentential logic:

(4) Phosphorus = Hesperus $\wedge \neg\Box$(Phosphorus = Hesperus),

(5) Hesperus = Hesperus $\wedge \Box$(Hesperus = Hesperus).

Applying (EG) to (4) and (5), we get:

(6) $\exists x(x =$ Hesperus $\wedge\neg\Box(x =$ Hesperus$))$,

(7) $\exists x(x =$ Hesperus $\wedge \Box(x =$ Hesperus$))$.

As Quine [92] points out, however, (6) and (7) are incompatible with interpreting $\forall x$ and $\exists x$ as objectual quantifiers meaning "for all objects x (in the domain D)" and "for at least one object x (in D)" and letting the identity sign stand for genuine identity between objects (in D). Because, under this interpretation, (6) and (7) imply that one and the same object, Hesperus, both is and is not necessarily identical with Hesperus, which seems absurd.

The following are classical proposals for solving Quine's interpretational challenge:

(i) *Russell–Smullyan* (Smullyan [99]). According to this proposal, all singular terms except variables are treated as *Russellian terms*, i.e., as "abbreviations" of definite descriptions that are eliminated from the language by means of contextual definition à la Russell. If we let 'Hesperus' and 'Phosphorus' be Russellian terms having minimal scope everywhere — which clearly corresponds to the intended reading — then the inference will not go through (i.e., once the Russell terms have been contextually eliminated): the (EG)-steps above will not correspond to valid steps in primitive notation. With this treatment of singular terms, the paradox is avoided. One has the feeling, however, that the problem has been circumvented rather than solved.

(ii) *Carnap* (at least the way Quine reads him): The individual variables are not taken to range over physical objects, but instead over individual concepts. According to this reading, the names 'Phosphorus' and 'Hesperus' stand for different but coextensive individual concepts. The identity sign is interpreted not as a genuine identity between physical objects but as coextensionality between individual concepts. That is, an identity statement '$u = v$' is true if and only if the terms 'u' and 'v' stand for coextensive individual concepts. According to this interpretation, (6) and (7) mean:

(6′) There is an individual concept x which actually coincides with the individual concept Hesperus but does not do so by analytical necessity.

(7′) There is an individual concept x which not only happens to coincide with the individual concept Hesperus but does so by analytic necessity.

No contradiction ensues from these two statements. The price for this interpretation, however, seems to be as Quine expresses it: "a curiously idealistic ontology which repudiates material objects".

1.4 The advent of possible worlds semantics

1.4.1 Semantics for quantified modal logic in 1957: Hintikka and Kanger

1957 was a pivotal year in the history of modal logic.[6] In that year Stig Kanger published his dissertation *Provability in Logic* and a number of other papers where he outlined a new model-theoretic semantics for quantified modal logic. In the same year, Jaakko Hintikka published two papers on the semantics of quantified modal logic: 'Modality as referential multiplicity' and 'Quantifiers in deontic logic' (Hintikka [39, 40]). There are some striking parallels between these works by Hintikka and Kanger, but there are also notable differences.

Hintikka and Kanger had both done important and closely similar work in non-modal predicate logic. Using so-called model sets (nowadays often called "Hintikka sets" or "downward saturated sets") for predicate logic, Hintikka [38] had developed a new complete and effective proof procedure for predicate logic.

Let $\mathcal{L}$ be a language of predicate logic with identity and let U be a non-empty set of individual constants that do not belong to $\mathcal{L}$. A *model set* (over U) is a set m of sentences of the expanded language $\mathcal{L}_U$ satisfying the following conditions:[7]

$(C.\neg)$	if $\neg\varphi \in m$, then $\varphi \notin m$,
$(C.\neg\neg)$	if $\neg\neg\varphi \in m$, then $\varphi \in m$,
$(C.\wedge)$	if $\varphi \wedge \psi \in m$, then $\varphi \in m$ and $\psi \in m$,
$(C.\neg\wedge)$	if $\neg(\varphi \wedge \psi) \in m$, then $\neg\varphi \in m$ or $\neg\psi \in m$,
$(C.\forall)$	if $\forall x\varphi \in m$, then for every constant a in U, $\varphi(a/x) \in m$,
$(C.\neg\forall)$	if $\neg\forall x\varphi \in m$, then for some constant a in U, $\neg\varphi(a/x) \in m$,
$(C.=)$	for no individual constant a in $\mathcal{L}_U, a \neq a \in m$,
$(C.\mathrm{Ind})$	if $\varphi(a/x) \in m$, where φ is atomic, and $a = b \in m$, then $\varphi(b/x) \in m$.

Hintikka showed, what nowadays goes under the name *Hintikka's lemma*, namely, that a set Γ of sentences is satisfiable (true in some Tarski-style model) iff it can be imbedded in a model set over some non-empty set U of (new) individual constants. Furthermore, he provided an effective proof procedure for classical predicate logic. The method is very similar to the nowadays more familiar semantic tableaux method of Beth [11].

Hintikka [38, p. 47] points out that there is a close connection between his proof procedure and proofs in Gentzen's sequent calculus. The systematic search for a counterexample of a formula φ corresponds to the backward application of the rules of Gentzen's

[6] See [24] for a comprehensive historical account of the development of possible worlds semantics. For a mathematical exposition of the development of modal logic, see [36]).

[7] Here we have assumed that $\neg$, $\wedge$ and $\forall$ are primitive and that $\vee$, $\rightarrow$ and $\exists$ are introduced as abbreviations in the usual way. For other choices of primitive logical constants, the definition of a model set has to be adjusted accordingly.

cut-free calculus for predicate logic. As a matter of fact, Kanger in *Provability in Logic* [53] provided an elegant effective proof procedure for classical predicate logic based on a sequent calculus that is equivalent to Hintikka's.

Hintikka's formal semantics for modal logic. When studying classical predicate logic, Hintikka and Kanger used strikingly similar techniques and obtained similar results. However, their approaches to modal logic were different. Kanger started out from the work of Tarski and set himself the task of extending the method of Tarski-style truth-definitions to predicate languages with modal operators. Hintikka, on the other hand, generalised his method of model sets to the case of modal logic. In doing so he invented the notion of a *model system*. Roughly speaking, a model system consists of a set Ω of model sets and a binary relation R defined between the members of Ω. Different versions of Hintikka's semantics impose different conditions on model sets, but in order simplify the exposition, we can say that a model system is an ordered pair $\mathcal{S} = \langle \Omega, R \rangle$, such that:

(a) Ω is a non-empty set of model sets for $\mathcal{L}$,

(b) R is a binary relation between the members of Ω (the alternativeness relation),

(c) for all $m \in \Omega$, if $\Box\varphi \in m$, then for all $n \in \Omega$ such that $mRn, \varphi \in n$,

(d) for all $m \in \Omega$, if $\neg\Box\varphi \in m$, then $\neg\varphi \in n$, for some $n \in \Omega$ such that mRn.

Hintikka thought of the members of Ω as partial descriptions of possible worlds. A set Γ of sentences is *satisfiable* (in the sense of Hintikka) iff there exists a model system $\mathcal{S} = \langle \Omega, R \rangle$ and a model set $m \in \Omega$ such that $\Gamma \subseteq m$. A sentence φ is valid iff the set $\{\neg\varphi\}$ is not satisfiable.

Hintikka [40] sketched a tableaux-style method of proving completeness theorems in modal logic. The idea is a generalisation of his proof procedure for first order logic. Hintikka [41] states (without formal proofs) that the systems **T**, **B**, **S4**, **S5** for sentential logic are sound and complete with respect to the Hintikka-style semantics where R is assumed to be reflexive, symmetric, reflexive and transitive and an equivalence relation, respectively. Rigorous completeness proofs using the tableaux method were published by Kripke, [64], for the case of quantified S5, and for numerous systems of propositional modal logic in [67, 68].[8]

An important difference between Hintikka's semantics for modal logic, on the one hand, and the ones developed by Carnap, Kanger and Montague [83], on the other, is that Hintikka allows the space of possibilities Ω to vary from one system to another. The only requirement is that Ω is a non-empty set satisfying the constraints (b), (c) and (d) above. In the formal semantics of Carnap, Kanger and Montague, on the other hand, the space of possibilities is fixed once and for all to be the set of all state descriptions (Carnap), the class of all systems (or alternatively, domains) (Kanger), or all first-order models over a given domain (Montague). One could say that Carnap, Kanger and Montague only allow interpretations of modalities that are in a sense *standard* and disallow *non-standard interpretations*. Thus, the relationship between Hintikka's semantics (and the one later developed by Kripke) and the ones developed by Carnap, Kanger and Montague is analogous to that between *standard* and *non-standard* semantics for higher-order

[8]In [65], Kripke announces a great number of completeness results in modal propositional logic. He also notes "For systems based on **S4**, **S5**, and **M**, similar work has been done independently and at an earlier date by K. J. J. Hintikka".

predicate logic. This distinction between the various approaches has been emphasised by Cocchiarella [23] and Hintikka [46]. Allowing non-standard interpretations for modal logics, of course, facilitated the proofs of completeness results, since the logics for logical or analytical necessity corresponding to the standard semantics are in general not recursively enumerable.

Kanger's Tarski-style semantics for quantified modal logic. Kanger's ambition was to provide a language of quantified modal logic with a model-theoretic semantics à la Tarski.[9]

A Tarski-style interpretation for a first-order predicate language $\mathcal{L}$ consists of a non-empty domain D and an assignment of appropriate extensions in D to every non-logical symbol and variable of $\mathcal{L}$. Kanger's basic idea was to relativise the notion of extension to various possible domains. In other words, he thought of an interpretation for a given language $\mathcal{L}$ as a *function* that *simultaneously* assigns extensions to the non-logical symbols and variables of $\mathcal{L}$ for *every* possible domain. Such a function Kanger called a (*primary*) *valuation*. Formally, a valuation for a language L of quantified modal logic is a function v which for *every* non-empty domain D assigns an appropriate extension in D to every individual constant, individual variable, and predicate constant in $\mathcal{L}$. Kanger also introduced the notion of a *system* $\mathcal{S} = \langle D, v\rangle$ consisting of a designated domain D and a valuation v. Notice that v does not only assign extensions to symbols relative to the designated domain D, but relative to *all* domains simultaneously.

Kanger then defined the notion of a formula φ being *true in a system* $\mathcal{S} = \langle D, v\rangle$ (in symbols, $\mathcal{S} \vDash \varphi$):

(1) $\mathcal{S} \vDash (t_1 = t_2)$ iff $v(D, t_1) = v(D, t_2)$,

(2) $\mathcal{S} \vDash P(t_1, \ldots, t_n)$ iff $\langle v(D, t_1), \ldots, v(D, t_n)\rangle \in v(D, P)$,

(3) $\mathcal{S} \nvDash \bot$,

(4) $\mathcal{S} \vDash (\varphi \to \psi)$ iff $\mathcal{S} \nvDash \varphi$ or $\mathcal{S} \vDash \psi$

(5) $\langle D, v\rangle \vDash \forall x\varphi$ iff $\langle D, v'\rangle \vDash \varphi$, for each v' such that $v' =_x v$,

(6) for every operator $\Box, \mathcal{S} \vDash \Box\varphi$ iff $\forall \mathcal{S}'$, if $\mathcal{S} R_\Box \mathcal{S}'$, then $\mathcal{S}' \vDash \varphi$.

Explanation: v' is like v except possibly at x (also written, $v' =_x v$) if and only if, for every domain U and every variable y other than $x, v'(U, y) = v(U, y)$. In the above definition, $R_\Box$ is a binary relation between systems that is associated with the modal operator $\Box$. $R_\Box$ is what is nowadays called the *accessibility relation* associated with the operator $\Box$. Kanger points out that by imposing certain formal requirements on the accessibility relation, like reflexivity, symmetry, transitivity, etc., one can make the operator satisfy corresponding well-known axioms of modal logic.

One source of inspiration for Kanger's use of accessibility relations in modal logic was no doubt the work of Jónsson and Tarski [52] on representation theorems for Boolean algebras with operators.[10] Jónsson and Tarski define operators $\Diamond$ on arbitrary subsets X of a set U in terms of binary relations $R \subseteq U \times U$ in the following way:

$$\Diamond X = \{x \in U : \exists y \in X(yRx)\},$$

[9] Cf. Kanger [53, 54, 55, 56, 57]). See also Lindström [81] for a more extensive discussion of Kanger's approach to quantified modal logic.

[10] On [53, p. 39] Kanger makes an explicit reference to Jónsson and Tarski [52].

that is $\Diamond X$ is the image of X under R. They also point to correspondences between properties of $\Diamond$ and properties of R. Among other things, they prove a representation theorem for so-called closure algebras that, via the Tarski-Lindenbaum construction, yields the completeness theorem for propositional **S4** with respect to Kripke models with a reflexive and transitive accessibility relation. However, Jónsson and Tarski do not say anything about the relevance of their work to modal logic.

Among the modal operators in $\mathcal{L}$, Kanger introduced two designated ones, **N** ("analytic necessity") and **L** ("logical necessity"), with the following semantic clauses:

$$\langle D, v\rangle \models \mathbf{N}\varphi \text{ iff for every domain } D', \langle D', v\rangle \models \varphi$$
$$\langle D, v\rangle \models \mathbf{L}\varphi \text{ iff for every system } \mathcal{S}, \mathcal{S} \models \varphi.$$

A formula φ is *true* in a system $\langle D, v\rangle$ iff $\langle D, v\rangle \models \varphi$. A formula φ is said to be *valid (logically true)* if it is true in every system $\langle D, v\rangle$. A formula φ is a *logical consequence* of a set Γ of formulæ (in symbols, $\Gamma \models \varphi$) if φ is true in every system in which all the formulæ in Γ are true.

In order to get a clearer understanding of Kanger's treatment of quantification, we shall speak of selection functions that pick out from each domain an element of that domain as *individual concepts*. We can think of a system $\mathcal{S} = \langle D, v\rangle$ as assigning to each individual constant c the individual concept $\{\langle D, v(D, c)\rangle : D \text{ is a domain}\}$ and to each variable x the individual concept $\{\langle D, v(D, x)\rangle : D \text{ is a domain}\}$. The formula $P(t_1, \ldots, t_n)$ is true in $\mathcal{S} = \langle D, v\rangle$ if and only if the individual concepts designated by $t_1, \ldots, t_n$ pick out objects in the domain D that stand in the relation $v(D, P)$ to each other. The identity symbol designates the relation of *coincidence* between individual concepts (at the "actual" domain D). That is, $t_1 = t_2$ is true in a system $\mathcal{S} = \langle D, v\rangle$ if and only if the individual concepts designated by t_1 and t_2, respectively, pick out one and the same object in the domain D of $\mathcal{S}$.

The universal quantifier $\forall x$ can now be thought of as an objectual quantifier that ranges not over the "individuals" in the "actual" domain D, but over the (constant) domain of all individual concepts. That is, $\forall x\varphi$ is true in a system $\langle D, v\rangle$ if and only if φ is true in every system that is exactly like $\langle D, v\rangle$ except, possibly, for the individual concept that it assigns to the variable x.

Kanger's solution to Quine's paradox of identity is essentially the same as Carnap's. Quine's objection to Kanger would therefore be the same as to Carnap: Kanger's quantifiers do not range over ordinary individuals but over individual concepts instead. Moreover, Kanger's treatment of quantification in modal contexts does not provide any means of *identifying* individuals from one domain to another. Hence there is no way of saying in Kanger's modal language that *one and the same* individual has a property P and possibly could have lacked P. That is, neither Carnap's nor Kanger's semantics can account for modality *de re*.

1.4.2 Hintikka's response to Quine's challenge

Quine's interpretational challenge seemed to place the advocates of quantified modal logic in a dilemma. They would either have to accept standard quantification theory (with the usual laws of universal instantiation, existential generalisation and indiscernibility of identicals) and reject quantified modal logic, or accept a quantified modal logic, where the quantifiers were interpreted in a non-standard way à la Carnap as ranging over

intensional entities (individual concepts), rather than over robust extensional entities as Quine would demand.

Hintikka [39, 40], however, rejected the terms in which Quine's interpretational challenge was stated. First of all he broadened the discussion by not only considering the logical modalities and Quine's metalinguistic interpretation of these, but also epistemic modalities ('It is known that φ') and deontic ones ('It is obligatory that φ'). He then introduced the idea of *referential multiplicity*. In answer to Quine's question whether a certain occurrence of a singular term in a modal context is purely referential, and thus open to substitution and existential generalisation, or non-referential, in which case substitution and existential generalisation would fail according to Quine, Hintikka [39] pointed to a third possibility. According to the classical Fregean approach [32] singular terms would in non-extensional contexts not have their standard reference but instead refer to intensional entities, their ordinary senses. Hintikka saw no need to postulate special intensional entities for the singular terms to refer to in non-extensional contexts. The failure of substitutivity was instead explained by the referential multiplicity of the singular terms and by the fact that in intensional contexts the reference of the terms in various alternative courses of events ("possible worlds") is considered simultaneously.

Informally Hintikka [39] expressed the basic ideas behind the possible worlds interpretation of modal logic in the following words:

> ...we often find it extremely useful to try to chart the different courses the events may take even if we don't know which one of the different charts we are ultimately going to make use of. ...This analogy is worth elaborating. The concern of a general staff is not limited to what there will actually be. Its business is not just to predict the course of a planned campaign, but rather to be prepared for all the contingencies that may crop up during it. ...Most of the maps prepared by the general staff represent situations that will never take place. ...There are for the most parts some actual units for which the marks on the map stand, and the mutual positions of the units are such that the situation could conceivably arise. ...But the location of the units on the maps may be different from the locations the units have or ever will have. Some of the marks may stand for units which have not yet been formed; other maps may be prepared for situations in which some of the existing units have been destroyed. All these features have their analogues in modal logic.

In this example Hintikka informally speaks of the same units as occurring in different situations ("cross-world identification of individuals") and of individuals coming into existence or disappearing as one goes from one situation to another ("varying domains").

Hintikka goes on to explain the bearing of the above example on referential opacity.

> We may perhaps say that when we are doing modal logic, we are doing more than one thing at one and the same time. We use certain symbols — constants and variables — to refer to the actually existing objects of our domain of discourse. But we are also using them to refer to the elements of certain other states of affairs that need not be realized. Or, which amounts to the same, we are employing these symbols to build up 'maps' or models for the purpose of sketching certain situations that will perhaps never take place. If we could confine our attention to one of these possible states of affairs at a time, the occurrences of our symbols would be purely referential. The

> interconnections between the different models interfere with this. But since the symbols are purely referential within each particular model, the deviation from pure referentiality is not strong enough to destroy the possibility of employing quantifiers with pretty much the same rules as in the ordinary quantification theory. If I had to characterize the situation briefly, I should say that the occurrences of our terms in modal contexts are not usually *purely referential*, but rather that they are *multiply referential*.

This idea of referential multiplicity is perhaps the basic intuitive idea behind the possible worlds interpretation of modal notions and of indexical semantics in general. It seems that Hintikka here gives one of the earliest, or perhaps the earliest, clear expression of the idea.

Hintikka's semantics for quantified modal logic is informally interpreted in such a way that the quantifiers range over genuine individuals. Thus, Hintikka has a notion of cross-world identification: one and the same individual may occur in different worlds. However, the semantics allows individuals to *split* from one world to another, i.e., the individuals a and b may be identical in one world w_0 but they may fail to be identical in some alternative world to w_0. Thus, the principle:

$$(\Box =) \quad \forall x \forall y(x = y \to \Box(x = y)), \quad (\textit{Necessity of Identity})$$

is not valid in Hintikka's semantics. As a consequence, the unrestricted principle of indiscernibility of identicals does not hold in modal contexts according to Hintikka (cf., Hintikka [41] and later writings).

Hintikka's solution to Quine's paradox of identity. There are two cases to consider:

(1) One or the other of the singular terms under consideration ('Hesperus' or 'Phosphorus') is not a "rigid designator", that is it does not designate the same individual in every possible world (or "scenario") under consideration. Then, existential generalisation fails and Quine's paradoxical argument does not go through.

(2) Each of the two names picks out "the same" individual in every world under consideration. However, some scenario w under consideration is such that the individual Hesperus in w is distinct from the individual Phosphorus in w. In this case, Quine's argument goes through, but Hintikka has to argue that the conclusion:

(6) $\exists x(x = \text{Hesperus} \land \neg\Box(x = \text{Hesperus}))$

(7) $\exists x(x = \text{Hesperus} \land \Box(x = \text{Hesperus}))$,

contrary to appearance, is not absurd, since an individual can "split" when we go from one possible scenario to one of its alternatives. Consider for example:

> Superman and Clark Kent are in fact identical, but Lois Lane doesn't believe that they are identical.

Hintikka may explain the apparent truth (according to the story) of this sentence by the fact that some scenarios (possible worlds) in which Superman and Clark Kent are different individuals are among Lois Lane's doxastic alternatives in the actual world (where they are identical).

1.4.3 Montague's early semantics for quantified modal logic

A semantic approach to first-order modal predicate logic that has a certain resemblance to Kanger's was developed by Montague [83].[11] Like Kanger, Montague starts out from the standard model-theoretic semantics for non-modal first-order languages and extends it to languages with modal operators. He defines an *interpretation* for an ordinary first-order predicate language $\mathcal{L}$ to be a triple $\mathcal{I} = \langle D, I, g\rangle$, where (i) D is a non-empty set (the *domain*); (ii) I is a function that assigns appropriate denotations in D to the non-logical constants (predicate symbols and individual constants) of $\mathcal{L}$; and (iii) a function g (an *assignment* in D) that assigns values in D to the individual variables of $\mathcal{L}$. For each non-logical constant or variable X, let $\mathcal{I}(X)$ be the *semantic value* (i.e., *denotation* for non-logical constants and *value* for variables) of X in the interpretation $\mathcal{I}$. Then the notion of *truth* relative $\mathcal{I}$ is defined as follows:

(1) $\mathcal{I} \vDash P(t_1, \ldots, t_n)$ iff $\langle \mathcal{I}(t_1), \ldots, \mathcal{I}(t_n)\rangle \in \mathcal{I}(P)$,

(2) $\mathcal{I} \vDash (t_1 = t_2)$ iff $I(t_1) = I(t_2)$,

(3) $\mathcal{I} \vDash \neg\varphi$ iff $\mathcal{I} \nvDash \varphi$,

(4) $\mathcal{I} \vDash (\varphi \to \psi)$ iff $\mathcal{I} \nvDash \varphi$ or $\mathcal{I} \vDash \psi$,

(5) $\mathcal{I} \vDash \forall x\varphi$ iff for every object $a \in D, \mathcal{I}(a/x) \vDash \varphi$.

Here, $\mathcal{I}(a/x)$ is the interpretation that is exactly like $\mathcal{I}$, except for assigning the object a to the variable x as its value.

Montague now asks the same question as Kanger: How can this definition of the truth-relation be generalised to first-order languages with modal operators? As we recall, Kanger solved the problem by modifying the notion of an interpretation: a Kanger-type interpretation (what he called 'a system') assigns denotations to the non-logical constants and values to the variables not only for one single domain (the 'actual' one) but for all domains in one fell swoop. Montague's approach is simpler than Kanger's: he keeps the notion of an interpretation $\mathcal{I}$ of first-order logic intact, and just adds semantic evaluation clauses for the modal operators. As in the Kanger semantics, each modal operator $\Box$ is associated with an accessibility relation $R_\Box$. Now, however accessibility relations are relations between interpretations $\mathcal{I} = \langle D, I, g\rangle$ of the underlying non-modal first-order language. The semantic clause corresponding to the operator $\Box$, with associated accessibility relation $R_\Box$, is:

(6) $\mathcal{I} \vDash \Box\varphi$ iff for every interpretation $\mathcal{I}'$ such that $\mathcal{I}R_\Box\mathcal{I}', \mathcal{I}' \vDash \varphi$.

Montague associates with the operator **L** of *logical necessity* the accessibility relation R_L defined by:

$$\langle D, I, g\rangle R_L \langle D', I', g'\rangle \text{ iff } D = D' \text{ and } g = g'.$$

Thus, his semantic clause for **L** becomes:

[11] Montague [83] writes: "The present paper was delivered before the Annual Spring Conference in Philosophy at the University of California, Los Angeles, in May, 1955. It contains no results of any great technical interest; I therefore did not initially plan to publish it. But some closely analogous, though not identical, ideas have recently been announced by Kanger [54, 55] and by Kripke in [64]. In view of this fact, together with the possibility of stimulating further research, it now seems not wholly inappropriate to publish my early contribution."

(7) $\langle D, I, g\rangle \vDash \mathbf{L}\varphi$ iff for every I' defined over D, $\langle D, I', g\rangle \vDash \varphi$.

That is, $\mathbf{L}\varphi$ is true in an interpretation $\mathcal{I}$ iff φ is true in every interpretation $\mathcal{I}'$ that is like $\mathcal{I}$ except for, possibly, assigning different semantic values to the non-logical constants of $\mathcal{L}$.

Stated in contemporary terms, Montague's semantic clause for the logical necessity operator becomes:

(8) $\mathbf{L}\varphi$ is true in a model $\mathcal{M} = \langle D, I\rangle$ relative to an assignment g iff for every model $\mathcal{M}'$ with domain D, φ is true in $\mathcal{M}'$ relative to g.

Let us say that a formula φ of $\mathcal{L}$ is *D-valid relative to g* iff for every model $\mathcal{M}$ with domain D, φ is true in $\mathcal{M}$ relative to g. We say that φ is *D-valid* iff it is D-valid relative to every assignment g in D. Then, from Montague's semantic clause for $\mathbf{L}$, we can conclude:

(9) $\mathbf{L}\varphi$ is true in $\mathcal{M} = \langle D, I\rangle$ relative to g iff φ is D-valid relative to g.

and

(10) $\mathbf{L}\varphi$ is true in $\mathcal{M} = \langle D, I\rangle$ iff φ is D-valid.

We say that a formula φ of $\mathcal{L}$ is *logically true* iff it is D-valid in every non-empty domain D.

Montague's [83] semantics for $\mathbf{L}$ is exactly what Cocchiarella [23] refers to as the "primary semantics" for logical necessity. Hence, we can reformulate Cocchiarella's [23] *incompleteness theorem* for that semantics as follows:

THEOREM 2. *Suppose that $\mathcal{L}$ contains at least one binary predicate symbol. Then, the set of logically true sentences in Montague's [83] semantics for logical necessity is not recursively enumerable. Thus, Montague's [83] logic for logical necessity is not axiomatizable.*

Montague's solution to Quine's paradox of identity. According to Montague's interpretation, $\mathbf{L}\varphi$ is logically equivalent with a *formula of second-order predicate logic* $()\varphi$, where $()$ stands for a string of universal quantifiers that bind all non-logical symbols in φ. In other words, Montague's semantics induces a translation from first-order modal logic to extensional second-order predicate logic. According to Montague's semantics from [83], the quantifier $\forall x$ is interpreted as a genuine quantifier over individuals. Free variables are "directly referential", i.e., a free variable is interpreted uniformly inside a formula as standing for one and the same individual regardless of where in the formula it occurs. Individual constants, on the other hand, are reinterpreted freely from one interpretation to another.

Montague's semantics validates the following principles without restrictions:

(LI) $\forall x(x = x)$, *(Law of Identity)*
(I=) $\forall x\forall y(x = y \rightarrow (\varphi(x/z) \rightarrow \varphi(y/z)))$. *(Indiscernibility of Identicals)*

In addition, we have: $\forall x\mathbf{L}(x = x)$. Therefore, the following principle is valid:

($\Box$I) $\forall x\forall y(x = y \rightarrow \mathbf{L}(x = y))$. *(Necessity of Identity)*

But the following is not valid:

$$\text{Phosphorus} = \text{Hesperus} \rightarrow \mathbf{L}(\text{Phosphorus} = \text{Hesperus}).$$

It follows that the principles of *Universal Specification* (US) and *Existential Generalisation* (EG) are not valid. Thus, Quine's paradoxical argument (Section 1.3, (1)–(7)) cannot be carried through within Montague's logic. Although (US) and (EG) cannot be applied to individual constants, they do hold for variables.

It appears that Montague's semantical interpretation satisfies all requirements imposed by Quine [92] on an interpretation of quantified modal logic for the logical modalities. However, Montague's semantics still has counterintuitive consequences. Consider, for instance, the following proof of the thesis that *everything there is exists necessarily*:

(1) $\forall x \exists y(x = y)$ predicate logic

(2) $\mathbf{L}\forall x \exists y(x = y)$ from (1) by necessitation

(3) $\forall x \exists y(x = y) \rightarrow \exists y(x = y)$ universal specification (US) (for variables)

(4) $\mathbf{L}(\forall x \exists y(x = y) \rightarrow \exists y(x = y))$ from (3) by necessitation

(5) $\mathbf{L}\exists y(x = y)$ from (2) and (4) by modal logic

(6) $\forall x \mathbf{L}\exists y(x = y)$ from (5) by universal generalization (UG)

This proof is valid according to Montague's semantics: line (1) is logically true and the steps in the proof preserve logical truth. It is also easy to see directly that the conclusion (6) of the argument is logically true according to Montague's definition. This conclusion, however, is extremely counterintuitive (provided we read the quantifiers in the normal way as ranging over ordinary objects). Intuitively, it is simply false that everything there is exists necessarily. Hence, there are still problems with Montague's semantics. We shall return to the above problematic argument in connection with Kripke's [66] possible worlds semantics.

It should also be noted that Montague's semantics validates the schema:

(I) $\exists x \mathbf{L}\varphi(x) \leftrightarrow \forall x \mathbf{L}\varphi(x)$.

i.e., φ holds necessarily of one thing just in case φ holds necessarily of everything. Moreover, the semantics validates the Barcan schema and its converse:

(BF) $\forall x \mathbf{L}\varphi(x) \rightarrow \mathbf{L}\forall x \varphi(x)$
(CBF) $\mathbf{L}\forall x \varphi(x) \rightarrow \forall x \mathbf{L}\varphi(x)$.

From (1), (BF) and (CBF) we infer:

(II) $\exists x \mathbf{L}\varphi(x) \leftrightarrow \mathbf{L}\forall x \varphi(x)$.

That is, a property holds necessarily of one thing just in case it is necessary that it holds of everything.

According to Montague's semantics the logically necessary properties are the same for everything; namely, just those properties that by logical necessity hold of everything. That is, Montague's semantics is *essentialist* in the weak Quinean sense of distinguishing between properties that hold necessarily of a thing and properties that hold only contingently of it. But it rejects the *strong essentialist thesis* that there are properties that some objects have necessarily and others do not have at all, or have only contingently

(cf. [8, 89]).[12] Hence, condition (I) seems to be correct, as long as we speak of logical necessity. Logic does not discriminate between individuals, so if F is a logically necessary property of one thing, it is a logically necessary property of everything there is.[13]

The Barcan formula and its converse, however, are dubious. Consider first (BF). Suppose that a is the only thing that exists. Then, $\forall x\mathbf{L}(x = a)$. However, it does not seem intuitively correct to infer: $\mathbf{L}\forall x(x = a)$. Next, consider (CBF). Clearly, $\mathbf{L}\forall x\exists y(x = y)$. If (CBF) were valid, we could infer $\forall x\mathbf{L}\exists y(x = y)$, which — as we have already pointed out — is counterintuitive. We will return to the semantic significance of (BF) and (CBF) in Section 1.4.4. Finally, condition (II) is clearly counterintuitive. Burgess [14] says of (II) that it "could silence any critic who claimed the notion of *de re* modality to be more obscure than that of *de dicto* modality, but would do so only at the cost of making *de re* notation pointless".

1.4.4 Kripke's semantics for quantified modal logic

Kripke 1959. The possible worlds semantics introduced by Kripke [64] may be cast in the following form (which differs from Kripke's original formulation in terminology as well as in some minor details). We consider a language $\mathcal{L}$ of modal predicate logic with identity containing for each $n \geq 1$, a denumerably infinite list of n-ary predicate symbols, but no function symbols or individual constants. Let D be a non-empty set. We define a *valuation* for $\mathcal{L}$ over D to be a function V which to every n-ary predicate symbol $P(n \geq 1)$ in $\mathcal{L}$ assigns a value $V(P) \subseteq D^n$. An *assignment* in D is a function g which to every individual variable x assigns a value $g(x) \in D$. A model over D is an ordered pair $\mathcal{M} = \langle \mathbf{K}, V_0 \rangle$ such that (i) $\mathbf{K}$ is a set of valuations for $\mathcal{L}$ over D, and (ii) $V_0 \in \mathbf{K}$.

Given a model $\mathcal{M} = \langle \mathbf{K}, V_0 \rangle$ over D, an evaluation V in $\mathbf{K}$, assignment g in D, and formula φ we define recursively what it means for φ to be *true in V relative to $\mathcal{M}$ and g* (in symbols: $V \vDash_M \varphi[g]$):

(1) $V \vDash_{\mathcal{M}} P(x_1, \ldots, x_n)[g]$ iff $\langle g(x_1), \ldots, g(x_n) \rangle \in V(P)$,

(2) $V \vDash_{\mathcal{M}} (x = y)[g]$ iff $g(x) = g(y)$,

(3) $V \vDash_{\mathcal{M}} \neg\varphi[g]$ iff $V \nvDash_{\mathcal{M}} \varphi[g]$,

(4) $V \vDash_{\mathcal{M}} (\varphi \to \psi)[g]$ iff $V \nvDash_{\mathcal{M}} \varphi[g]$ or $V \vDash_{\mathcal{M}} \psi[g]$,

(5) $V \vDash_{\mathcal{M}} \forall x\varphi[g]$ iff for every object $a \in D, V \vDash_{\mathcal{M}} \varphi[g(a/x)]$,

(6) $V \vDash_M \Box\varphi$ iff for every valuation V' in $\mathbf{K}$, $V' \vDash_{\mathcal{M}} \varphi$.

As usual, $g(a/x)$ is the assignment that is exactly like g except for assigning a to the variable x.

[12]See also Kaplan's [61] penetrating analysis of the distinction between logical and metaphysical necessity. According to Kaplan, logical necessity is committed to a *benign* form of Aristotelian essentialism that "makes a specification of an individual essential only if it is logically true of that individual". Metaphysical necessity, on the other hand, is *invidious*, since it allows for distinct individuals to have different essential properties.

[13]On the other hand, (I) is clearly counterintuitive for metaphysical necessity. Let, for example, $\varphi(x)$ be the formula '$(\exists y(y = x) \to x \in \{\text{Socrates}\})$' and let $\Box$ stand for metaphysical necessity. Then, $\Box\varphi(\text{Socrates})$ is true. Socrates is a member of {Socrates}, in every possible world where Socrates exists. But, of course, $\Box\varphi(\text{Plato})$ is false. Thus (I) fails for metaphysical necessity.

We say that φ is *true in $\mathcal{M}$ relative to g* if $V_0 \vDash_{\mathcal{M}} \varphi[g]$. φ is *true in $\mathcal{M}$* if $V_0 \vDash_M \varphi[g]$ for every assignment g in D. φ is *valid in the domain D* if φ is true in all models $\mathcal{M}$ over D. φ is *universally valid* if φ is valid in every non-empty domain D (i.e., just in case φ is true in every model $\mathcal{M}$).

Kripke gives the following intuitive motivation for this semantics: The valuations in **K** are thought of as representing the set of all "possible" (or "conceivable" or "imaginable") worlds. The valuation V_0 represents the "real" world. It is assumed that the set D of individuals is the same for all possible worlds. Necessity is defined as truth in all possible worlds.

Kripke's [64] semantics validates all the classically valid schemata of first-order predicate logic with identity, the characteristic axioms of **S5**, as well as the Barcan formula (BF) and its converse (CBF). The set of valid sentences is closed under modus ponens, uniform substitution, necessitation, and universal generalization. In [64], Kripke defines a formal system $\mathbf{S5}^{*=}$ for quantified modal logic and proves using semantic tableaux methods that it is sound and complete for the given semantics.

Let us now compare Kripke's [64] semantics with Montague's semantics [83] for logical necessity. Let us say that a Kripke [64] model $\mathcal{M} = \langle \mathbf{K}, V_0 \rangle$ over a non-empty domain D is *maximal* if **K** contains all valuations for $\mathcal{L}$ over D.[14]

Montague's semantics for logical necessity differs from Kripke's [64] semantics in considering maximal models only. We obtain Montague's semantics for logical necessity by imposing the requirement on Kripke's [64] models that the set **K** should contain all valuations V for $\mathcal{L}$ over D. Hence, a sentence φ of $\mathcal{L}$ is logically true in Montague's [83] semantics for logical necessity iff it is true in all maximal Kripke [64] models. By restricting our attention to maximal models, we get what Cocchiarella [23] calls the "primary semantics" for logical necessity.

At this point it is natural to ask what intended interpretation Kripke had in mind for the necessity operator in 1959. Was it logical necessity, analytical necessity, or perhaps some kind of metaphysical necessity? One reason for thinking that Kripke's notion of necessity in 1959 was not logical necessity is his use of models that are non-maximal (or "non-standard" in the terminology of Hintikka [46]). Instead of working with all models or valuations over D, like Montague, or with all possible systems as Kanger, Kripke is considering an arbitrary non-empty subset of all possible valuations. This feature of his models may suggest that Kripke's intended interpretation of the necessity operator is not strict logical necessity, but perhaps instead some kind of metaphysical necessity. This conclusion is however, not unavoidable: Kripke's intended interpretation of the necessity operator could still have been logical necessity and his *intended interpretations* could still be some or all of the *maximal models*. Kripke's reason for allowing non-maximal models, in addition to maximal ones, when defining validity, could have been logical rather then philosophical.[15] If Kripke, like Kanger and Montague, had chosen to work only with maximal models, the set of valid sentences would not have been recursively enumerable and there would be no completeness theorem to be proved. Kripke's intended model could, for instance, be a maximal model over some infinite set. A modal sentence of an interpreted language of modal predicate logic would then be *true* if it was true in the

[14] The term "maximal model" was introduced by Parsons [89] in connection with Kripke's [66] semantics for quantified logic. It is less tendentious than Hintikka's term "standard model".

[15] Ballarin [4] argues that Kripke's development of his possible worlds semantics was driven entirely "by formal considerations, not interpretive concerns".

intended model. Interpreted in this way, Kripke's 1959 approach would be very close to Montague's of 1960. The only essential difference would be Kripke's use of non-standard models in addition to the standard ones for the purpose of defining a notion of universal validity that is recursively enumerable.

On the other hand, in [64, p. 3], Kripke speaks of **K** as representing the set of all "conceivable" worlds. He writes "... a proposition $\Box B$ is evaluated as true when and only when B holds in all conceivable worlds". This seems to indicate that Kripke's operator $\Box$ of [1959] should not be interpreted as strict logical necessity. It is very likely that the set of valuations representing all "conceivable" worlds is a proper subset of the set of absolutely all valuations. Thus Kripke may have had philosophical reasons, in addition to formal ones, for favouring a "non-standard" semantics allowing non-maximal models to a "standard" one.[16]

Kripke 1963. We present a version of Kripke's [66] semantics for modal predicate logic with identity, where the notion of a possible world is an explicit ingredient of the semantic theory. We differ from Kripke [66] in letting the language $\mathcal{L}$ contain individual constants.

A (*Kripke*) *frame* (or to use Kripke's own terminology, a *model structure*) for a language $\mathcal{L}$ of first-order modal predicate logic (with identity and individual constants, but no function symbols) is a quintuple $\mathcal{F} = \langle W, D, R, E, w_0\rangle$ where, (i) W is a non-empty set; (ii) D is a non-empty set; (iii) $R \subseteq W \times W$; (iv) E is a function which to each $w \in W$ assigns a subset E_w of D; and (v) w_0 is a designated element of W. Intuitively we think of matters thus: W is the set of all (*possible*) *worlds* (possible states of affairs, possible ways the world could have been), D is the set of all (*possible*) *individuals*, R is the *accessibility relation* between worlds, for each world w, E_w is the set of *individuals that exist in* w; and w_0 is the *actual world*. It is required that $D = \bigcup_{w\in W} E_w$, i. e., that every possible individual exists in at least one world.

Next, let us say that I is an *interpretation* (in D with respect to W) if it is a family of functions I_w, where w ranges over W, such that I_w assigns a subset $I_w(P)$ of D^n to each n-ary predicate constant P of $\mathcal{L}$ and an element $I_w(c) \in D$ to each individual constant c of $\mathcal{L}$. A *Kripke model* (for $\mathcal{L}$) is an ordered pair $\mathcal{M} = \langle \mathcal{F}, I\rangle$, where $\mathcal{F} = \langle W, D, R, E, w_0\rangle$ is a frame and I is an interpretation in D with respect to W. A model $\mathcal{M}$ of the form $\langle \mathcal{F}, I\rangle$ is said to be *based on* the frame $\mathcal{F}$.

Observe that $I_w(P)$ is not necessarily a subset of $(E_w)^n$, i. e., the extension of P in w may contain individuals that do not exist in w. Nor do we require that $I_w(c) \in E_w$. An *assignment* in $\mathcal{M}$ is a function g which assigns to each variable x an element $g(x)$ in D. For any term t in $\mathcal{L}$, we define $\mathcal{M}_w(t, g)$ to be $g(t)$ if t is a variable; and $I_w(t)$ if t is an individual constant. We speak of $\mathcal{M}_w(t, g)$ as the *denotation of the term t at the world w relative to the model $\mathcal{M}$ and the assignment g.*

With these notions in place, we can define what it means for a formula φ to be *true at a world w with respect to the model $\mathcal{M}$ and the assignment g* (in symbols, $w \vDash_{\mathcal{M}} \varphi[g]$):

(1) $w \vDash_{\mathcal{M}} P(t_1, \ldots, t_n)[g]$ iff $\langle \mathcal{M}_w(t_1, g), \ldots, \mathcal{M}_w(t_n, g)\rangle \in I_w(P)$.

(2) $w \vDash_{\mathcal{M}} (t_1 = t_2)[g]$ iff $\mathcal{M}_w(t_1, g) = \mathcal{M}_w(t_2, g)$.

(3) $w \vDash_{\mathcal{M}} \neg\varphi[g]$ iff $w \nvDash_{\mathcal{M}} \varphi[g]$.

[16] Cf., however, Almog [1, p. 217], who writes about Kripke [64]: "... Kripke had at the time nothing more than "complete assignments," and the modality he worked with was definitely *logical* possibility".

(4) $w \vDash_{\mathcal{M}} (\varphi \rightarrow \psi)[g]$ iff $w \nvDash_{\mathcal{M}} \varphi[g]$ or $w \vDash_{\mathcal{M}} \psi[g]$.

(5) $w \vDash_{\mathcal{M}} \forall x\varphi[g]$ iff, for every $a \in E_w$, $w \vDash_{\mathcal{M}} \varphi[g(a/x)]$.

(6) $w \vDash_{\mathcal{M}} \Box\varphi[g]$ iff, for every $u \in W$ such that $wRu, u \vDash_{\mathcal{M}} \varphi[g]$.

We say that φ is *true with respect to the model $\mathcal{M}$ and the assignment g* (in symbols $\vDash_{\mathcal{M}} \varphi[g]$), iff φ is true at the actual world w_0 with respect to $\mathcal{M}$ and g. φ is *true in the model $\mathcal{M}$* (in symbols, $\vDash_{\mathcal{M}} \varphi$), if for every assignment $g, \vDash_{\mathcal{M}} \varphi[g]$. φ is *true in a frame $\mathcal{F}$* (in symbols, $\vDash_{\mathcal{F}} \varphi$) if φ is true in every model based on $\mathcal{F}$. Let $\mathbf{K}$ be a class of frames. We say that φ is $\mathbf{K}$*-valid* if φ is true in every $\mathcal{F} \in \mathbf{K}$.

Observe that there are two notions of validity that are naturally defined on classes of Kripke frames. With respect to the notion that we have just defined — we may call it *real-world validity* — the actual world plays a special role: a sentence φ is real-world valid in a class $\mathbf{K}$ of frames if it is true at the actual world in every frame in $\mathbf{K}$. Then, there is another notion of validity that we may call *general validity*: A sentence φ is generally valid in a class $\mathbf{K}$ just in case it is true at each world w in each frame in $\mathbf{K}$.[17] In the definition of general validity, the designated point of a Kripke model does not play any role. Thus, if we are only interested in general validity, there is no need to provide Kripke frames with designated worlds. Let us write $\vDash_{\mathbf{K}}$ and $\vDash^*_{\mathbf{K}}$ for real-world validity in $\mathbf{K}$ and general validity in $\mathbf{K}$, respectively. Then we have, for any sentence φ of $\mathcal{L}$

(1) $\vDash^*_{\mathbf{K}} \varphi$ iff $\vDash_{\mathbf{K}} \Box\varphi$

Let us say that a class $\mathbf{K}$ of Kripke frames is *normal* iff it satisfies the condition:

> Whenever $\mathcal{F}$ is in $\mathbf{K}$ and $\mathcal{F}'$ is a frame that differs from $\mathcal{F}$ only with respect to which world is the actual one, then $\mathcal{F}'$ is also in $\mathbf{K}$.

For normal classes of frames, real-world validity coincides with the general validity. Thus, for any sentence φ of $\mathcal{L}$,

(1) if $\mathbf{K}$ is normal, then $\vDash_{\mathbf{K}} \varphi$ iff $\vDash^*_{\mathbf{K}} \varphi$

The semantic import of the Barcan formula and its converse. Notice that Kripke frames in general have *varying domains*, i.e., the domains of quantification E_w are allowed to vary from one possible world to another. We say that a frame $\mathcal{F} = \langle W, D, R, E, w_0 \rangle$ has *increasing domains* iff for all $u, v \in W$, if uRv, then $E_u \subseteq E_v$. $\mathcal{F}$ has *decreasing domains* iff for all $u, v \in W$, if uRv, then $E_v \subseteq E_u$. $\mathcal{F}$ has *locally constant domains* iff for all $u, v \in W$, if uRv, then $E_u = E_v$. $\mathcal{F}$ has *globally constant domains* iff for all $u \in W, E_u = D$. We also say that F is a *constant domain frame* iff $\mathcal{F}$ has globally constant domains.

Consider now the following conditions on frames $\mathcal{F}$:

(ID) $\mathcal{F}$ has increasing domains.

(DD) $\mathcal{F}$ has decreasing domains.

(LCD) $\mathcal{F}$ has locally constant domains.

[17] Cf. [51, 22–24], for a comparison between the two concepts of logical truth (validity) and for the history of the distinction between the two.

(CBF) Every instance of the converse Barcan formula: $\Box\forall x\varphi(x) \to \forall x\Box\varphi(x)$, is generally valid in every model based on $\mathcal{F}$.

(BF) Every instance of the Barcan formula: $\forall x\Box\varphi(x) \to \Box\forall x\varphi(x)$, is generally valid in every model based on $\mathcal{F}$.

(CBF + BF) Every instance of the Barcan formula and its converse is generally valid in every model based on $\mathcal{F}$.

There is an exact correspondence between the conditions (ID), (DD), (LCD) and (CBF), (BF) and (CBF + BF), respectively (cf. [30]). That is:

(i) $\mathcal{F}$ has increasing domains iff it satisfies (CBF).

(ii) $\mathcal{F}$ has decreasing domains iff it satisfies (BF).

(iii) $\mathcal{F}$ has locally constant domains iff it satisfies (CBF + BF).

Moreover,

(iv) A sentence is generally valid in the class of all constant domain frames iff it is generally valid in all locally constant domain frames.

We may introduce an *existence predicate* **E** as a new logical constant and give it the semantic clause:

$$w \vDash_{\mathcal{M}} \mathbf{E}(t)[g] \text{ iff } \mathcal{M}_w(t,g) \in E_w.$$

However, this is unnecessary as long as we have identity in the language, since the predicate **E** is definable in terms of the existential quantifier and identity:

$$w \vDash_{\mathcal{M}} \mathbf{E}(t)[g] \text{ iff } w \vDash_{\mathcal{M}} \exists y(y = t)[g], \text{ where } y \text{ is a variable that is distinct from } t.$$

Hence, we may take $\mathbf{E}(t)$ as an abbreviation of $\exists y(y = t)$.

In terms of **E** we can express the requirements of increasing and decreasing domains in a simple way:

(v) $\mathcal{F}$ has increasing domains iff the sentence $\Box\forall x\Box\mathbf{E}(x)$ is valid in $\mathcal{F}$.

(vi) $\mathcal{F}$ has decreasing domains iff the formula $\Box(\Diamond\mathbf{E}(x) \to \mathbf{E}(x))$ is valid in $\mathcal{F}$.

We are especially interested in frames where R is the *universal relation* in W, i.e., in which:

$$w \vDash_{\mathcal{M}} \Box\varphi[g] \text{ iff, for every } u \in W, u \vDash_{\mathcal{M}} \varphi[g].$$

Let **QS5=** be the class of all such frames. It follows from what we have stated above, that neither the Barcan formula nor its converse is (**QS5=**)-valid.

In order to illustrate the difference between Kripke's [66] semantics and his earlier semantics from 1959, consider again the purported proof that *everything there is exists necessarily* (Section 1.4.3). The proof is valid in the semantics of Montague [83] as well as in Kripke [64]. However, according to Kripke [66], the argument fails. It is easy to see that the conclusion is not valid according to Kripke [66]. When we look at the purported proof, we see that it is line (3) that fails:

(3) $\forall x \exists y(x = y) \rightarrow \exists y(x = y)$ universal specification (US) (for variables)

That is, (US) is not valid according to Kripke [66] (not even for variables): The universal quantifier in the antecedent of (3) ranges over the domain of actually existing objects, while the free variable x in the succedent may take possible objects as values that lie outside the domain of actually existing objects. The failure of this intuitively invalid argument in Kripke's [66] semantics speaks in favour of this semantics in comparison with Montague [83] and Kripke [64].

Rigid designators. Kripke's [66] semantics validates the *Law of Identity*,

(L=) $\forall x(x = x)$,

as well as the principle of *Indiscernibility of Identicals*,

(I=) $\forall x \forall y[x = y \rightarrow (\varphi(x/z) \rightarrow \varphi(y/z))]$,

applicable without restrictions also to modal contexts $\varphi(z)$. From these principles, together with the rule of Necessitation it is easy to infer:

($\Box =$) $\forall x \forall y(x = y \rightarrow \Box(x = y))$ *(Necessity of Identity)*

($\Box \neq$) $\forall x \forall y(x \neq y \rightarrow \Box(x \neq y))$. *(Necessity of Distinctness)*

However, neither

(1) $c = d \rightarrow \Box(c = d)$

nor

(2) $c \neq d \rightarrow \Box(c \neq d)$,

is valid, for arbitrary individual constants c, d. This reflects an important difference between how individual variables and individual constants are treated in our modelling: in spite of their name, the denotation of individual constants may vary from one possible world to another, whereas the denotation of variables — in spite of their name — remains fixed throughout the universe of possible worlds. Here is obviously a niche to be filled! Suppose we introduce a new syntactic category of *names* and require that the interpretation of a name $\mathbf{n}$ be constant over the set of all possible worlds in any model $\mathcal{M}$; formally,

$$I_u(\mathbf{n}) = I_v(\mathbf{n}),$$

for all $u, v \in W$. Then, if $\mathbf{n}$ and $\mathbf{m}$ are any names, then:

(3) $\mathbf{n} = \mathbf{m} \rightarrow \Box(\mathbf{n} = \mathbf{m})$

(4) $\mathbf{n} \neq \mathbf{m} \rightarrow \Box(\mathbf{n} \neq \mathbf{m})$.

are both valid. The proposed modification amounts to treating the elements of the new category of names as what is now known, after Kripke [71], as *rigid* designators. In [71] Kripke made the claim that ordinary "proper names" in natural language are rigid designators.

Maximal models and maximal validity. Next, we introduce a special kind of Kripke models that we refer to as *maximal models*. We say that an ordered triple $\langle D, A, V \rangle$ is a

first-order model for $\mathcal{L}$ with *outer domain* D and *inner domain* A iff (i) $D \neq \varnothing, A \subseteq D$; and (ii) for each n-ary predicate constant $P, V(P) \subseteq D^n$; (iii) for each individual constant $c, V(c) \in D$.

A Kripke model $\mathcal{M} = \langle W, D, R, E, w_0, I\rangle$ is *maximal* if (i) $R = W \times W$; (ii) for every subset A of D and every first-order model $\langle D, A, V\rangle$ with outer domain D and inner domain A, there exists a $w \in W$ such that $E_w = A$ and $I_w = V$; and (iii) if $u, v \in W$ and $E_u = E_v$ and $I_u = I_v$, then $u = v$. Thus, in a maximal Kripke model with individual domain D, the possible worlds can be identified with all first-order models with outer domain D. Thus, for each non-empty set D, there is a unique maximal Kripke model with individual domain D.

The notion a maximal Kripke model is due to Terence Parsons [89]. Montague's [83] models correspond to the maximal Kripke models with a constant domain, i.e. where each $E_w = D$. If $\mathcal{M}$ is the maximal Kripke model with domain D, then for every formula φ of $\mathcal{L}$:

> $\Box\varphi$ is true at a world w in $\mathcal{M}$ relative an assignment g iff φ is true in every first-order model with outer domain D relative to g.

Thus, it is natural to interpret $\Box$ as a kind of logical (or combinatorial) necessity with respect to maximal Kripke models: $\Box\varphi$ is true in a maximal model with domain D iff φ is true in every first-order model with outer domain D.

Let us say that a formula φ is *maximally valid* iff for every maximal Kripke model $\mathcal{M}$ and every assignment g in $\mathcal{M}, \vDash_M \varphi[g]$. Observe that the set of maximally valid sentences is not closed under uniform substitution of arbitrary sentences for atomic sentences: for an atomic formula $Pc, \Diamond Pc$ is maximally valid, but, of course, $\Diamond\varphi$ is not in general maximally valid. Moreover, if φ is a formula that does not contain $\Box$ or $\Diamond$ which is not a theorem of first-order logic, then $\neg\Box\varphi$ is maximally valid. Of course, neither the Barcan schema nor its converse is maximally valid.

Suppose now that the intended model of $\mathcal{L}$ is some maximal Kripke model $\mathcal{M}_0$ with an infinite domain D_0. Then, all sentences of the form:

$$(n)\ \ \Diamond\exists x_1 \ldots \exists x_n(x_1 \neq x_2 \wedge \ldots \wedge x_1 \neq x_n \wedge x_2 \neq x_3 \wedge \ldots \wedge x_2 \neq x_n \wedge \ldots \wedge x_{n-1} \neq x_n),$$

where $x_1, \ldots, x_n$ are $n (n > 1)$ distinct variables, are *true* in (the intended model for) $\mathcal{L}$. This appears to be as it should be, given the interpretation of $\Diamond$ as (a kind of) logical possibility. With this notion of truth in $\mathcal{L}$, we can associate various notions of *logical truth*. One alternative is to say that a sentence in $\mathcal{L}$ is logically true iff it is true in every maximal model with the given outer domain D. With this notion all the sentences (n) come out as logically true. Another alternative is to say that a sentence is logically true if it is maximally valid, i.e., true in all maximal Kripke models. Then the sentences (n) are no longer logically true. Finally, we may identify logical truth in $\mathcal{L}$ with truth in all **QS5=**-Kripke models. Of these choices, only the last one satisfies the standard requirement on a logic of being closed under uniform substitution. Thus, if we insist that a logic should be closed under uniform substitution, it is reasonable to identify logical truth in $\mathcal{L}$ with Kripke's notion of universal validity. Hence, regardless of whether the intended model is a maximal model or not, we may reasonably conclude that the logic of alethic necessity is the set of all **QS5**=-valid sentences. By this line of reasoning, we come to the conclusion that regardless of whether we interpret $\Box$ as standing for logical

or metaphysical necessity, the *logic* of $\Box$ will be the same.[18]

Kripke versus Quine. In 1959 Kripke wrote:

> It is noteworthy that the theorems of this paper can be formalized in a metalanguage (such as Zermelo set theory) which is "extensional," both in the sense of possessing set-theoretic axioms of extensionality *and* in the sense of postulating no sentential connectives other than the truth-functions. Thus it is seen that at least a certain non-trivial portion of the semantics of modality is available to an extensionalist logician.

Perhaps, Kripke meant that he had refuted Quine's scepticism about quantified modal logic. Had he not after all done for quantified modal logic what Tarski and others had done for non-modal predicate logic: provided it with an extensional set-theoretic semantics? In addition he had axiomatised the logic and proved it complete for the given semantics. What else could one require of the interpretation of a logic?

Quine, however, was not satisfied. In 1972 he writes in a review of Kripke's paper 'Identity and Necessity' [96]:

> The notion of possible world did indeed contribute to the semantics of modal logic, and it behoves us to recognize the nature of its contribution: it led to Kripke's precocious and significant theory of models of modal logic. Models afford consistency proofs; also they have heuristic value; but they do not constitute explication. Models, however clear in themselves, may leave us still at a loss for the primary, intended interpretation.

Whatever was his aim in 1959 or 1963, in his later work Kripke's project is not to give an explanation of modal concepts in non-modal terms. In the Preface to *Naming and Necessity*, 1980 he writes:

> I do not think of 'possible worlds' as providing a *reductive* analysis in any philosophically significant sense, that is, as uncovering the ultimate nature, from either an epistemological or a metaphysical point of view, of modal operators, propositions, etc., or as 'explicating' them.

Clearly, Kripke's essentialist concept of necessity ("metaphysical necessity") simply cannot be reductively explained in non-modal terms.

Among other modellings for predicate modal logic, David Lewis's counterpart theory should be mentioned.[19] According to the Kripke paradigm, an individual may exist in more than one possible world (with respect to our formal modelling, it is possible that E_u and E_v should overlap, even if $u \neq v$). For Lewis, however, each individual inhabits only its own possible world; but it may have counterparts in other possible worlds. This approach has also been influential, both in philosophical and in mathematical quarters.

1.5 General intensional logic

1.5.1 Carnap-Montague's Intensional Logic

Frege's theory of *Sinn* (*sense*) and *Bedeutung* (*denotation*, *reference*), which was outlined in the article 'Über Sinn und Bedeutung' [32] has great intuitive appeal. In particular,

[18] Cf. [15].
[19] Cf. [75, 37].

it seems to provide elegant and intuitively appealing solutions to the familiar difficulties concerning:

(i) the cognitive significance of identity statements: how can '$a = b$' if true, be an informative statement differing in cognitive significance from '$a = a$'?

(ii) the problem of oblique or non-extensional contexts: how can two meaningful expressions with the same denotation (extension) ever fail to be interchangeable *salva veritate*?

(iii) the problem of providing an adequate truth-conditional semantics for propositional attitude reports.

Fregean solutions to these problems essentially involve the distinction between sense and denotation. The appearance of oblique contexts in natural languages was interpreted by Frege as indicating a certain kind of systematic ambiguity rather than a failure of extensionality. According to Frege's doctrine of indirect denotation, expressions denote in (unembedded) oblique contexts what is ordinarily their sense. Frege's extensional point of view has been advocated and developed in the 20th century by Alonzo Church [19, 20, 21] in his *Logic of Sense and Denotation*.[20]

Carnap [17], although still working within the Fregean tradition, saw the occurrence of oblique contexts in natural languages as genuine counterexamples to the *principle of extensionality*, according to which the denotation of a meaningful expression is always a function of the denotations of its semantically relevant parts.

According to Carnap [17], each well-formed expression of a language has both an *extension* (corresponding to Frege's denotation) and an *intension* (roughly corresponding to Frege's sense). Intuitively, the intension of a sentence is the proposition that the sentence expresses and the extension is the truth-value (true or false) of the sentence. A proposition partitions the set of all possible worlds in two cells: (i) the set of all worlds in which the proposition is true; and (ii) the set of all worlds in which the proposition is false. Carnap, therefore, proposed to identify a proposition p with the function f_p from the set W of all possible worlds to truth-values which for every possible world w has the value $f_p(w) =$ the truth-value of p in the world w. Thus, propositions are identified with functions from possible worlds (or in Carnap's case, from state descriptions, or set-theoretical models, that are taken to represent possible worlds) to truth-values. The *intension* of a sentence is the proposition it expresses and its *extension* in a possible world w is the truth-value in w of the proposition it expresses.

The intension of a predicate expression is intuitively the property (or relation-in-intension) that the predicate expresses. A property of individuals determines for every possible world w, the set of individuals that has the property in that world. Hence, a *property* P, can according to Carnap and Montague be identified with a function f_P from the set W of all possible worlds to sets of individuals, which for every possible world w

[20]As emphasised by Church [22] and Kaplan [60], the Fregean tradition in intensional logic should be distinguished from the quite different tradition stemming from Russell where the sense/denotation distinction is avoided. Russellian semantics, in contrast to Fregean semantics, assigns only one kind of semantic value, most naturally thought of as a kind of denotation, to the well-formed expressions of a language. In Russellian semantics, (logically) proper names refer (directly) to objects, sentences designate Russellian propositions, i.e. complexes of properties and objects, and predicates stand for propositional functions. Modern so-called theories of direct reference belong to the Russellian tradition (cf., for instance, [98]).

has the value $f_P(w)$ = the set of all entities that in the world w has the property P. For instance, the property of being red, is identified with the function form possible worlds to individuals that associates with every possible world the set of red objects in that world. Similarly, an n-ary *relation-in-intension* R is identified with a function from possible worlds to sets of ordered n-tuples. The intension of a predicate expression is the property or relation-in-intension it expresses and its extension in a possible world w is the set or relation-in-extension that is the value of that intension in the world w.

Finally, singular terms have individuals as their extensions and their intensions are what Carnap calls *individual concepts*, i.e., functions from possible worlds to individuals. The singular term 'the Greek philosopher that taught Alexander the Great' has in the actual world Aristotle as its extension. In another possible world, the extension may be Plato. In possible worlds where there is no unique Greek philosopher that taught Alexander, the singular term might be assigned an arbitrary conventional extension, the *null extension.* Since proper names, presumably, are *rigid designators* (cf. [71]) they have the same extension in every possible world (or at least in every possible world where the bearer of the name exists). Hence, the intension of a proper name is a constant function picking out the same object in every possible world (or at least this is the case for rigid designators of objects that exist necessarily, for instance, the numerals designating the natural numbers). On Kripke's view, co-referring proper names have the same intension. As a result, if a and b are co-referring proper names, then '$a = a$' and '$a = b$' have the same intension. Thus, it seems that difference in cognitive significance cannot be explained by difference in intension.

Kripke's [66, 67, 68] major innovation was his use — within each model structure — of a set of abstract points (indices, "possible worlds") to represent the space of possibilities. This innovation made it possible for Montague [84] — building on ideas from Carnap [17] — to represent intensional entities (senses, intensions) by set-theoretic functions from points (representing possible worlds) to extensions. Every kind of meaningful expression has according to Carnap-Montague semantics a suitable *intension*, i.e., a function from possible worlds to appropriate extensions. If E is an expression with intension $Int(E)$, and w is a possible world, then $Int(E)(w)$, i.e., the result of applying the intension of E to the possible world w, is the *extension of E in the world w* (in symbols $Ext_w(E)$). The *extension of E*, $Ext(E)$, is the extension of E in the actual world.

Following Carnap [17] we distinguish between different kind of constructions (or contexts) Φ:

(i) Φ is *extensional* iff there exists a function f_Φ such that for every possible world w, and all (appropriate) expressions $E_1, \dots, E_n$, $Ext_w(\Phi(E_1, \dots, E_n)) = f_\Phi(Ext_w(E_1), \dots, Ext_w(E_n))$. An *extensional language* is a language where every grammatical construction is extensional. An extensional language satisfies the *principle of extensionality*, i.e., the principle that the extension of a complex expression is always a function of the extensions of its semantically meaningful constituents.

(ii) Φ is *intensional* iff there exists a function F_Φ such that for all (appropriate) expressions $E_1, \dots, E_n$, $Int(\Phi(E_1, \dots, E_n)) = F_\Phi(Int(E_1), \dots, Int(E_n))$. An intensional language is a language in which every grammatical construction is intensional. Intensional languages satisfy the *principle if intensionality*, i.e., the principle that the intension of a complex expression is always a function of the intensions of its semantically meaningful constituents.

The principles of extensionality and intensionality are special cases of the *principle of compositionality*, i.e., the principle that the meaning of a complex expression is determined by its structure and the meaning of its constituents (cf., [104]).

The classical Boolean connectives are, of course, paradigm examples of extensional constructions. By modifying the above definitions slightly, in order to take variable binding operators into account, the classical quantifiers $\forall$ and $\exists$ are naturally construed as extensional operators as well. The modal operators $\Box$ and $\Diamond$, on the other hand, are examples of constructions that are intensional but not extensional. Carnap also considered propositional attitude constructions like 'John believes that ... ', that in his opinion were not even intensional. Such constructions for which the principle of intensionality fails, may be called *ultraintensional*.

In order to give a semantic analysis of belief contexts, Carnap introduced the notion of *intensional isomorphism* [17, §14]. Roughly speaking, two expressions are intensionally isomorphic iff they are built up from atomic expressions with the same intensions in the same way. Intensionally isomorphic expressions were said to have the same *intensional structure*. The intensional structure of an expression can thus be identified with the equivalence class of all expressions of the given language that are intensionally isomorphic with it. Intensional isomorphism and intensional structure was Carnap's explications of the intuitive notions of synonymy and meaning, respectively.[21] The intensional structures that correspond to sentences may be viewed as *structured propositions* in contrast to Carnapian propositions (functions from possible worlds to truth-values) that lack syntactical structure.[22] Carnap suggested that belief and other propositional attitudes be operators on such structured propositions rather than on intensions. If so, then intensionally isomorphic expressions are substitutable *salva veritate* in propositional attitude contexts. This seems fairly reasonable since one might argue that synonymous expressions are substitutable in such contexts.

Montague's intensional logic **IL** is a typed λ-calculus.[23] There are two basic types e and t of (possible) *individuals* and *truth-values* (true and false), respectively. Then, there is for every two types α and β, a type $(\alpha\beta)$ of *functions* from entities of type α to entities of type β. Finally, for every type α, there is a type $(s\alpha)$ of *senses* appropriate for entities of type α. Montague identifies the senses with Carnapian intensions, i.e., the members of $(s\alpha)$ are functions from possible worlds to entities of type α. All the domains of the various types are constant from one world to another. In particular, there is one domain of individuals that is common to all possible worlds. Thus, the domain of individuals is best thought of as the domain of all *possible* individuals.

For every type α, the language of **IL** contains variables and non-logical constants of type α. It also contains the logical constants: $=$ (identity), λ (lambda-abstraction), $\hat{}$ (intensional abstraction), $\check{}$ (intensional application), and brackets [,]. The sentential connectives, quantifiers $\forall$, $\exists$, and modal operators $\Box, \Diamond$, are definable in terms of $=, \lambda$, $\hat{}$, and $\check{}$ (Gallin [33, 15-16]). For each type α, one can quantify in IL over all the entities of type α. In particular, one can quantify over the collection of all *possible individuals.*

---

[21] This theme is developed further in Lewis [76].

[22] See King [63] for an overview of more recent work on structured propositions and references to the relevant literature (including work by David Kaplan, Nathan Salmon, Scott Soames, Jeff King, and others within the "direct reference"-tradition on so-called "Russellian propositions").

[23] See Montague [84], and especially Gallin [33] for a thorough model-theoretic study of Montague's intensional logic. In particular, Gallin presents an axiomatisation of Montague's intensional logic and proves that it is strongly complete with respect to general Henkin-type models.

In other words, **IL** is committed to an ontology of possible individuals.

Complex terms of **IL** are built up from atomic terms (variables and constants as follows): (i) If A is a term of type $(\alpha\beta)$ and B is a term of type α, then $[AB]$ is a term of type β; (ii) If A is a term of type β and x is a variable of type α then λxA is a term of type $(\alpha\beta)$: (iii) If A, B are terms of the same type, then $[A = B]$ is a term of type t; (iv) If A is a term of type α, then $\hat{}A$ is a term of type (sα); (v) If A is a term of type (sα), then $\check{}A$ is a term of type α. Terms of type t are called *formulæ*.

In the semantics, every (closed) term A of type α is assigned an extension $Ext_w(A)$ of type α relative to w, for each possible world w. The intension $Int(A)$ of A is then the function from worlds to extensions such that for each w, $Int(A)(w) = Ext_w(A)$. For each term A, $\hat{}A$ is a name of the intension of A. And, for each term A denoting an intension F, $\check{}A$ is a term which at every world w, refers to the value of F at w. Hence, $(A = \check{}\,\hat{}A)$ will always hold. The semantics of **IL** satisfies the principle of intensionality and $\hat{}$ is the only primitive non-extensional construction of **IL**. The modal operator $\Box$ is defined in **IL** as follows:

$$\Box\varphi =_{df.} [\hat{}\varphi = \hat{}T],$$

that is, φ is necessarily true iff the intension of φ equals the intension of any tautology T. $\Box$ is an **S5**-operator and the Barcan formulæ and their converses are valid in the semantics.

Montague's intensional logic admits quantifying into intensional constructions. According to Montague's intended interpretation, the individual quantifiers range over *possible* individuals. Quantification over actual individuals can be analysed by means of the introduction of an existence predicate. However, Montague's use of quantifiers ranging over possibilia is of course an abomination in the eyes of Quine and likeminded philosophers who favour an actualist metaphysics.

1.5.2 Church's logic of sense and denotation

The expressions of natural language are according to the Fregean view *systematically ambiguous*: both the sense and the denotation of an expression vary with the linguistic context in which it occurs. This systematic ambiguity is the basis for Church's program [19, 20, 21] of representing natural language discourse involving oblique contexts within a formal language the logic of which is completely *extensional*, that is, in which the ordinary principles of substitutivity of classical logic are valid. His fundamental idea is to let each expression A of the natural language be represented by different expressions $A_0, A_1, A_2, \ldots$ of the formal language depending on the context in which A occurs. Suppose, for instance, that the sentence "Tom is married", when it occurs in a non-oblique context, is translated as **Married(Tom)**. Then, the sentence (1), where the verb phrase "suspects that" gives rise to an oblique context, may be represented as:

(2) **Suspects**(**Mary**, **Married**$_1$(**Tom**$_1$)),

where **Married**$_1$ and **Tom**$_1$ are atomic expressions that denote the (ordinary) senses of **Married** and **Tom**, respectively. Analogously,

(3) George knows that Mary suspects that Tom is married

may be represented as

(4) **Knows**(**George**, $\mathbf{Suspects}_1(\mathbf{Mary}_1, \mathbf{Married}_2(\mathbf{Tom}_2))$).

Using Church's terminology, we may say that $\mathbf{Tom}_1$ and $\mathbf{Tom}_2$ denote *the concept of being Tom* and *the concept of being the concept of being Tom,* respectively. In this way ambiguity is avoided in the representing language and the classical principles of substitutivity as well as all other principles of classical logic are preserved.

Church's logic of sense and denotation is a simple type theory that has much in common with Montague's intensional logic **IL** but which differs from **IL** in not violating the principle of extensionality. In Montague's language there is, as we recall, only one non-extensional operator $\hat{}$ which transforms a term A into a term $\hat{}A$ that denotes the intension of A. Since A occurs in $\hat{}A$, $\hat{}$ is non-extensional. Church's logic of sense and denotation, on the other hand, is fully extensional. For each denoting expression A, there is in Church's language another expression $\langle A \rangle$, denoting the sense of A. Since $\langle A \rangle$ does not contain A as a syntactic part, the occurrence of A in the language does not violate extensionality. $\langle A \rangle$ replaces A in oblique contexts. For instance, the indirect discourse construction: 'John believes that φ' is replaced by the direct discourse version: 'John believes $\langle \varphi \rangle$', where $\langle \varphi \rangle$ is a name of the proposition expressed by the sentence φ. The construction 'John believes $\langle \varphi \rangle$' differs from 'John believes $\hat{}\varphi$' in being fully extensional.

In Church [18] and [19], three alternative principles of individuation for senses were proposed referred to as Alternatives (0), (1) and (2). The alternative that individuates senses most coarsely is Alternative (2), according to which two expressions have the same sense iff they are logically equivalent. Roughly speaking, Alternative (2) amounts to identifying Fregean senses with Carnapian intensions, i.e., with functions from possible worlds (or models or state descriptions representing possible worlds) to denotations (or extensions). Thus, Alternative (2) is the alternative which is closest to modern possible worlds semantics.

The alternative that is closest to Frege's own conception of sense is probably Alternative (0), according to which two terms A and B have the same sense, if and only if they are *intensionally isomorphic* in the sense of Carnap [17]. In addition to alternatives (0) and (2), Church also considered an intermediate alternative called Alternative (1), which is fairly close to Alternative (0) but seems to have less intuitive motivation. According to Alternative (1) expressions that are lambda-convertible to each other have the same sense.

Church's logic of sense and denotation is not directly concerned with linguistic expressions and their senses and denotations, but rather with the language-independent so-called *concept relation* that holds between senses and the entities that they are senses of. As Church points out in [21], the more finely senses are individuated, the more closely will the abstract theory of senses and their objects resemble the more concrete theory of names and their denotations, with the concept relation playing a role similar to the one played by the *denotation predicate* of semantics. Consequently, antinomies analogous to the semantic antinomies may arise for formulations of the logic of sense and denotation along the lines of Alternative (0) or (1). Indeed, Myhill [85] points out that Church's Alternative (0) is threatened by the antinomy described by Russell in *The Principles of Mathematics*, Appendix B, p. 527, the so-called *Russell-Myhill paradox* (cf. Anderson [2]).

The development of a logic of sense and denotation along the lines of Alternative (0) — taking Carnap's intensional isomorphism, Church's synonymous isomorphism, or some related notion as a criterion for two expressions having the same sense — is of great

theoretical interest. First of all, the fundamental principle of Alternative (0):

$$sense(FA) = sense(FB) \rightarrow sense(A) = sense(B),$$

seems to be involved whenever a difference in sense between FA and FB is *explained* in terms of a difference in sense between A and B. Secondly, any principle of individuation for senses that is substantially less strict than Alternative (0) seems to be inadequate for a Fregean treatment of the logic of propositional attitudes.

Unfortunately, however, the attempts so far to develop a logic of sense and denotation along the lines of Alternative (0) have led either to inconsistency or to great complications, for instance, in the form of an infinite hierarchy of concept relations of different orders. Furthermore, no entirely satisfactory explanation has so far been given of the notion of sense involved in Alternative (0). Related to this is the lack of an intuitive semantic theory for Alternative (0) and a corresponding notion of logical validity.

However, the pursuit of Church's Alternative (2) has made considerable progress. Thus, David Kaplan [58, 60] and Charles Parsons [88] have provided versions of Church's logic of sense and denotation with a possible worlds semantics of Carnap-Montague type. Parsons [88] even shows that his version of Church's logic of sense and denotation is exactly equivalent to (intertranslatable with) Montague's intensional logic. Moreover he provides an axiomatisation of Church's Alternative 2 that is equivalent to Gallin's axiomatisation of Montague's intensional logic.

1.6 Logical and metaphysical necessity

We make a rough distinction between two types of intuitive interpretations of the operators $\Diamond$ and $\Box$ of alethic modal logic. First there is the *metaphysical* or *counterfactual* interpretation:

$\Diamond\varphi$: either φ, or it could have been the case that φ.
$\Box\varphi$: φ, and it could not have been the case that $\neg\varphi$.

Then, there is the *logical* or *metalogical* interpretation:

$\Diamond\varphi$: it is not self-contradictory to assume that φ is the case.
$\Box\varphi$: it is self-contradictory to assume that $\neg\varphi$ is the case.

From now on, we shall use $\mathbf{L}\varphi$ and $\mathbf{M}\varphi$ for the logical modalities and reserve $\Box$ and $\Diamond$ for the metaphysical ones.

According to the *possible worlds analysis* of metaphysical necessity:

$\Box\varphi$ is true at a possible world w iff φ is true at every possible world.

There is an extensive and fast growing philosophical literature on the proper analysis of the notion of a possible world (cf. [25, 87]). Roughly speaking, we are distinguishing between *the world* as the (concrete) totality of everything there is and *possible worlds* as total alternative ways the world could have been (cf. [71, pp. 15–20]). Characterised in this way, possible worlds are abstract entities: *total possible states of the world.* This notion of possible world should be contrasted with David Lewis's notion of a possible world as a concrete alternative universe (cf. [80]). Regardless of our ultimate understanding of possible worlds, to say that a statement φ is *true at* a possible world w means, intuitively, that φ, with its actual meaning, would have been true (simpliciter) had w obtained.

A delicate question that now arises is how metaphysical necessity relates to logical necessity. The answer, of course, depends on how precisely we characterise the notion of logical necessity. Different semantic characterisations give rise to different answers. Suppose that we define logical necessity in terms of a class K of (admissible) models (or interpretations). Each model $\mathcal{M}$ is associated with a set $U_{\mathcal{M}}$ of points (representing "possible worlds") of which one is the designated point $@_{\mathcal{M}}$ (representing "the actual world"). We write $u \vDash_{\mathcal{M}} \varphi$ for the sentence φ being *true at the point u in the model $\mathcal{M}$*. *Truth in a model $\mathcal{M}$* is defined as truth at the designated point $@_{\mathcal{M}}$ of the model $\mathcal{M}$. *Logical truth*, or *validity*, is defined as truth in every model in K. We assume that:

(i) $u \vDash_{\mathcal{M}} \neg\varphi$ iff not: $u \vDash_{\mathcal{M}} \varphi$

(ii) $u \vDash_{\mathcal{M}} (\varphi \rightarrow \psi)$ iff either $u \nvDash_{\mathcal{M}} \varphi$ or $u \vDash_{\mathcal{M}} \psi$.

(iii) $u \vDash_{\mathcal{M}} \mathbf{L}\varphi$ iff for every model $\mathcal{N}$ in $K, @_{\mathcal{N}} \vDash_{\mathcal{N}} \varphi$.

(iv) $u \vDash_{\mathcal{M}} \Box\varphi$ iff for every point $v \in U_{\mathcal{M}}, v \vDash_{\mathcal{M}} \varphi$.

Given this type of semantics, there is no guarantee that logical necessity implies metaphysical necessity. Suppose, for example, that the language contains a logical constant **actually** with the semantic clause:

(v) $u \vDash_{\mathcal{M}}$ **actually** (φ) iff $@_{\mathcal{M}} \vDash_{\mathcal{M}} \varphi$,

i.e., **actually** (φ) is true at a point in a model iff φ is true at the designated point in the model. Then, every instance of the following schema is valid:

(1) $\mathbf{L}(\varphi \leftrightarrow$ **actually** $(\varphi))$,

although, the following schema fails (in both directions):

(2) $\Box(\varphi \leftrightarrow$ **actually** $(\varphi))$.

We can easily construct models $\mathcal{M}$ for a sentential language of the indicated kind for which (2) fails.

Thus it appears, as Zalta [108] has argued, that logical necessity does not imply metaphysical necessity. There are logical truths that are metaphysically contingent. However, this claim is highly counterintuitive. There are various ways of avoiding the conclusion that logical truth does not imply metaphysical necessity. One may, for one reason or another, refuse constructions like **actually**, that make reference to special worlds, the status of logical constants.

Another option is to modify the notion of logical truth. The notion of logical truth that we have employed is the one we have called *real-world validity*. It is the notion according to which a statement φ is logically true (valid) iff it is true at the actual world in each model. As we have seen, however, there is an alternative notion, *general validity*, according to which a statement is logically true iff it is true at each world in each model.

Let us write $\vDash$ and $\vDash^*$ for real-world validity and general validity, respectively. The two notions are related as follows: For any statement φ,

(1) $\vDash \varphi$ iff $\vDash^*$ **actually** (φ).

(2) $\vDash^* \varphi$ iff $\vDash \Box\varphi$.

The operator **L** was introduced by "reflecting" the meta-linguistic notion of real-world validity into the object language. We can also introduce an operator $\mathbf{L}^*$ corresponding to the notion of general validity. The semantic clauses for **L** (*real-world logical necessity*) and $\mathbf{L}^*$ (*general logical necessity*) are:

(vi) $u \vDash_{\mathcal{M}} \mathbf{L}\varphi$ iff for every model $\mathcal{N}$ in $K, @_{\mathcal{N}} \vDash_{\mathcal{N}} \varphi$.

(vii) $u \vDash_{\mathcal{M}} \mathbf{L}^*\varphi$ iff for every model $\mathcal{N}$ in K and every point v in $\mathcal{N}, v \vDash_{\mathcal{N}} \varphi$.

That is, **L** corresponds to truth at the actual world in each model and $\mathbf{L}^*$ corresponds to truth at every world in each model. The two notions of logical necessity are interdefinable:

(1) $\vDash^* \mathbf{L}\varphi \leftrightarrow \mathbf{L}^*\mathbf{actually}(\varphi)$.

(2) $\vDash^* \mathbf{L}^*\varphi \leftrightarrow \mathbf{L}\Box\varphi$.

Moreover, we have:

(3) $\vDash^* \mathbf{L}^*\varphi \rightarrow \Box\varphi$,

although, as we have seen, the corresponding implication does not hold for real-world logical necessity, i.e., for **L**.

Metaphysical necessity does not imply logical necessity. It does not appear self-contradictory to think, as the Greeks did, that water is an element. But since water, as it turned out, is a compound of oxygen and hydrogen, it could not have been an element. There is, so to speak, no counterfactual situation, or possible world, where *water* is not a compound. So even if it is not logically necessary, it is metaphysically necessary that water is a compound. Hence, the statement:

(1) Water is a compound

is metaphysically necessary (assuming that "water", is a rigid designator), but it is not logically necessary. In conclusion, we can say that real-world logical necessity (**L**) neither implies nor is implied by metaphysical necessity ($\Box$). General logical necessity ($\mathbf{L}^*$) on the other hand, implies metaphysical necessity, but is not implied by it.

The (epistemological) distinction between *a priori* and *a posteriori* also comes in here. In Kripke's theory, (1) exemplifies a statement that, although metaphysically necessary, is nevertheless *a posteriori*. On the other hand, given certain assumptions, "The Paris meter is one meter long" may be an example of a statement that is true *a priori* but is not metaphysically necessary [71].

2 THE MODAL LOGIC OF BELIEF CHANGE

In this section, modal logic is brought to bear on an area which has already reached a degree of maturity (although still in need of further development) and which has been formulated with little or no regard to modal logic. By re-formulating the theory in terms of modal logic, a degree of systematisation is gained, and — it is hoped! — the theoretical understanding of the theory is enhanced.

2.1 Introduction

2.1.1 Two paradigms

The theory of belief change is a fairly sprawling phenomenon. In the tradition examined here, one is interested in how new information is handled by a "rational" agent. By assumption, the agent is situated in an environment, often referred to as "the world". The world is always in some world state or other, and the agent is always in some belief state or other. From time to time, the agent is presented with new information about the world. The problem is to describe how the new information affects the current belief state. In the two cases studied here, two further assumptions are made: that the new information is always accepted, and that acceptance always leads to a uniquely defined (usually, but not necessarily, different) belief state.

We distinguish between two cases: the case when the world is static (the world does not change) and the case when the world is dynamic (the world might change); belief change is called *belief revision* in the former case, *belief update* in the latter. We also distinguish between two attitudes which an agent may have and which are called *conditionalisation* and *imaging*, respectively; these terms, which have an origin in probability theory, will not be explained (see Lewis [128], Gärdenfors [112]). The two paradigms selected for study here, AGM and KM, exemplify those two attitudes: AGM is a conditionalising and KM is an imaging theory. It is commonly accepted that a conditionalising attitude is appropriate for belief revision and an imaging attitude for update. Thus AGM is said to be a theory of belief revision and KM a theory of update.

In this section we shall offer explications within modal logic of both the AGM paradigm and the KM paradigm. They are not meant to be exact counterparts of AGM and KM as they were historically defined; they are rather meant to bring out what we take to be essential to those conceptions. (Our use of the terms "AGM" and "KM" is ambiguous: they stand both for certain people (Alchourrón, Gärdenfors and Makinson in the former case, Katsuno and Mendelzon in the latter) and for the theories developed by those authors.)

2.1.2 Revision

There is a strong connection between the theory of belief change and the logic of conditionals. That this should be so is not so surprising if it is remembered that the following much quoted passage in a paper of Frank Ramsey, published posthumously, inspired both fields:

> If two people are arguing 'If p will q?' and are both in doubt as to p, they are adding p hypothetically to their stock of knowledge and arguing on that basis about q; so that in a sense 'If p, q' and 'If $p, {\sim}q$' are contradictories. We can say that they are fixing their degrees of belief in q given p. If p turns out false, these degrees of belief are rendered *void*. If either party believes p for certain, the question ceases to mean anything for him except as a question about what follows from certain laws or hypothesis. [133, p. 149].

Both Robert Stalnaker and David Lewis cite this passage as a point of departure for their respective theories of conditional logic (Stalnaker [140], Lewis [127]). But it was read also by Peter Gärdenfors, who was looking for a different kind of theory of conditionals, one

with a semantics not formulated in terms of possible worlds. Given a formal language of some familiar sort, what an agent believes on a certain occasion may be represented by what Gärdenfors called a *belief set*, namely, the set of propositions believed by the agent. It is assumed that belief sets are theories in Tarski's sense, that is, that they contains all tautologies and are closed under modus ponens. Fundamental for Gärdenfors's theory of belief change is the existence of an operation $*$ such that, for any T, if T is an agent's belief set on a certain occasion and φ is a proposition, then $T * \varphi$ is the agent's belief set if and after he has revised his beliefs by φ. Given $*$, Ramsey may be read as suggesting that two people, who share a belief set T and argue 'If φ will ψ?', can be represented as arguing whether ψ is an element of $T * \varphi$. In this way Gärdenfors was led to look for *assertability conditions* for conditionals rather than *truth conditions*. In particular, he had hoped to find a conditional $\Rrightarrow$ satisfying the following form of the so-called Ramsey Test:

$$(RT) \qquad \varphi \Rrightarrow \psi \in T \text{ iff } \psi \in T * \varphi.$$

Everything now hangs on the properties of the revision operation $*$. Proceeding in the same way as C. I. Lewis when the latter was trying to characterise his modal operators, Gärdenfors laid down a number of postulates in order to characterise $*$. Let K be a certain *background theory*, that is, a special belief set that is taken for granted and not subject to revision. A *K-theory* is a theory that includes K. We say that a formula φ is *K-consistent* if the set $K \cup \{\varphi\}$ is K-consistent, that a formula φ is *K-consistent with* a belief set T if $T \cup \{\varphi\}$ is consistent, and that two formulæ φ and ψ are *K-equivalent* if $\varphi \leftrightarrow \psi \in K$. For any set Σ, we write $Cn(\Sigma)$ for the set of tautological consequences of Σ. In the following postulates, T is supposed to be a K-theory.

(AGM1) For any formula $\varphi, T * \varphi$ is a K-theory.

(AGM2) $\varphi \in T * \varphi$

(AGM3) $T * \varphi \subseteq Cn(T \cup \{\varphi\})$.

(AGM4) φ is K-consistent with T, then $Cn(T \cup \{\varphi\}) \subseteq T * \varphi$

(AGM5) If φ is K-consistent, then $T * \varphi$ is K-consistent.

(AGM6) If φ and ψ are K-equivalent, then $T * \varphi = T * \psi$.

(AGM7) $T * (\varphi \wedge \psi) \subseteq Cn(T * \varphi \cup \{\psi\})$.

(AGM8) If ψ is K-consistent with $T * \varphi$, then $Cn(T * \varphi \cup \{\psi\}) \subseteq T * (\varphi \wedge \psi)$.

Some of these postulates have received their own names in the literature: (AGM2) is the Success Postulate, (AGM4) the Preservation Postulate and (AGM5) the Consistency Postulate.

This, in a nutshell, is the syntactic side of AGM, the famous paradigm created by Gärdenfors in collaboration with Carlos Alchourrón and David Makinson [109, 113]. Now it turns out, as Gärdenfors himself was the first to observe, that if the condition (RT) is added to the AGM-postulates (after the new operator $\Rrightarrow$ has been added to the object language), then the result is, not inconsistency, but triviality. This is yet another interesting example of how intuitions, which on the face of it seem quite reasonable, turn out jointly to be incompatible. But it is also a wonderful example of the old

saying that they who seek will find, but not always what they are looking for. For even though Gärdenfors did not find his conditional, he did find, together with Alchourrón and Makinson, a seminal theory of belief revision.

2.1.3 Update

A different theory of belief change is given in Katsuno and Mendelzon [122] (see also Grahne [115]). They emphasise a distinction, which they attribute to Keller and Winslett [123], between *knowledge-adding changes* (revisions) and *change-recording updates*. According to Katsuno and Mendelzon, we believe that the real world is one of a certain set of possible worlds, which one we may not know. When we are informed that the real world has changed in a certain respect, we examine each of the old possible worlds and ask how our beliefs would have changed if that particular one had been the real world (notice the counterfactual!). "The fact the real world has changed gives us no grounds to conclude that some of the old worlds were actually not possible." (This is the feature that made us classify Katsuno and Mendelzon's theory as imaging.)

Where AGM have belief sets, KM have *knowledge bases* (abbreviated KBs), each knowledge base consisting of just one formula ("since we need a finite fixed representation of a KB to store it in a computer"). Like AGM, KM also introduce a new operator: if φ is a KB and ψ is a formula (intuitively, the new information) then $\varphi\Diamond\psi$ is the KB that results from updating φ with ψ. Assuming a propositional language with only finitely many letters, they propose the following postulates:

(KM1) $\varphi\Diamond\psi$ implies ψ.

(KM2) If φ implies ψ, then $\varphi\Diamond\psi$ is equivalent to φ.

(KM3) If both φ and ψ are satisfiable, then $\varphi\Diamond\psi$ is also satisfiable.

(KM4) If φ is equivalent to φ' and ψ is equivalent to ψ' then $\varphi\Diamond\psi$ is equivalent to $\varphi'\Diamond\psi'$.

(KM5) $(\varphi\Diamond\psi) \wedge \theta$ implies $\varphi\Diamond(\psi \wedge \theta)$.

(KM6) If $\varphi\Diamond\psi$ implies ψ' and $\varphi\Diamond\psi'$ implies ψ, then $\varphi\Diamond\psi$ is equivalent to $\varphi\Diamond\psi'$.

(KM7) If φ is such that, for all χ, φ implies χ or φ implies $\neg\chi$, then $(\varphi\Diamond\psi) \wedge (\varphi\Diamond\psi')$ implies $\varphi\Diamond(\psi \vee \psi')$.

(KM8) $(\varphi \vee \varphi')\Diamond\psi$ is equivalent to $(\varphi\Diamond\psi) \vee (\varphi'\Diamond\psi)$.

An important difference between the two paradigms is that, while the AGM operator $*$ is not part of the language in which the agent's beliefs are expressed, the KM operator $\Diamond$ is. For this reason, AGM is not a logic, in the usual sense of the word, but KM is.

2.1.4 Translations

To a modal logician, it is obvious that AGM can be re-formulated as a modal logic. A Rosetta stone with inscriptions in ordinary language, AGM language and the language of modal logic might contain the following text:

ordinary language	AGM	modal logic
the agent believes that φ	$\varphi \in T$	$\mathbf{B}\varphi$
after revising his beliefs by φ, the agent believes that χ	$\chi \in T * \varphi$	$[*\varphi]\mathbf{B}\chi$

What is called modal logic here is of course doxastic logic enriched with change operators loaned from dynamic logic. In particular, if φ is a formula, then the expression $*\varphi$ functions like a term that denotes 'the agent's acceptance of φ' — an event, possibly an action. In this way the nonformal theory of AGM can be translated, more or less faithfully, into a formal theory within what we shall call dynamic doxastic logic (DDL); details are provided below. (We prefer "doxastic" to "epistemic" since belief need not be veridical.)

To give a direct translation of KM is more difficult. The KM-operator $\Diamond$ (a binary operator not to be confused with the homonymous unary higher-order operator appearing in DDL-operators of type $[\Diamond\varphi]$) is a propositional connective but not one of classical logic. A KM-formula

$(*)$ $(\varphi \Diamond \psi) \to \chi$

might at first sight seem to represent the claim that if an agent, who believes that φ, updates his beliefs by ψ, then he will believe, after the update, that χ. If so, then the DDL-formula $\mathbf{B}\varphi \to [\Diamond\psi]\mathbf{B}\chi$ would be a natural translation of $(*)$. However, there is a great difference between the total of an agent's beliefs — a knowledge base, to use KM-terminology — and a single belief of the agent. Therefore a faithful translation into DDL requires a strengthening of our current language. One possibility would be to adopt an operator $\mathbf{E}$ of a kind first considered by Hector Levesque, $\mathbf{E}\varphi$ carrying the intuitive meaning "the agent believes exactly that φ (and what follows logically from φ)" or "all that the agent believes is that φ (and what follows logically from φ)". In this more expressive language

$$\mathbf{E}\varphi \to [\Diamond\psi]\mathbf{B}\chi$$

would be an adequate translation of $(*)$. Unfortunately, Levesque's operator is not easy to axiomatise (see [124]).

2.1.5 Some object languages

A number of object languages will figure in the sequel, and it is a good idea to give careful definitions of them at this point. Let LETT be a denumerable set of letters. We assume a truth-functionally complete supply of *boolean* operators, *conditional* operators $\sqsupset$ and $>$, *doxastic* operators $\mathbf{B}$, $\mathbf{b}$, $\mathbf{K}$, and $\mathbf{k}$, as well as a *star operator* $*$, a *rhombus operator* $\Diamond$, and *change operators* $[\ ]$ and $\langle\ \rangle$. The operators $\mathbf{B}$, $\mathbf{K}$ and $[\ ]$ are so-called box operators, while the operators $\mathbf{b}$, $\mathbf{k}$ and $\langle\ \rangle$ are dual so-called diamond operators; for simplicity, in what follows we shall treat the latter as abbreviatory devices: that is, for all appropriate formulæ χ, $\mathbf{b}\chi =_{\mathrm{df}} \neg\mathbf{B}\neg\chi$, $\mathbf{k}\chi =_{\mathrm{df}} \neg\mathbf{K}\neg\chi$, $\mathbf{b}\chi =_{\mathrm{df}} \neg\mathbf{B}\neg\chi$, $\langle *\varphi\rangle\chi =_{\mathrm{df}} \neg[*\varphi]\neg\chi$ and $\langle\Diamond\varphi\rangle\chi =_{\mathrm{df}} \neg[\Diamond\varphi]\neg\chi$. In the same way, we also stipulate that $\varphi > \psi =_{\mathrm{df}} \neg(\varphi \sqsupset \neg\psi)$.

Informally, formulæ of type $\varphi \sqsupset \psi$ and $\varphi > \psi$ may be read as "if φ then ψ" or, if a distinction between them is called for, as "if φ then certainly ψ" and "if φ then conceivably ψ". (But we are not committed to any particular reading, be it "ontic", "epistemic", "dynamic", or whatever.) The operators $\mathbf{B}$ and $\mathbf{K}$ are for belief and for doxastic commitment, respectively. For many purposes a reading of laziness of $\mathbf{K}\varphi$ as

"the agent knows that φ" is all right, but "the agent is doxastically committed to φ" is better, for implicit in the theories we are considering below is the assumption that what is referred to is implied by a certain, usually not specified, sometimes possibly variable, background theory. The change operators are the after-operators of dynamic logic.

Classical language

$\textsc{lett} \subseteq \textsc{boole}$,

$\varphi, \psi \in \textsc{boole} \Rightarrow (\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \to \psi), (\varphi \leftrightarrow \psi), \neg\varphi \in \textsc{boole}$.

Basic doxastic language

$\textsc{boole} \subseteq \textsc{basic}{\cdot}\textsc{dox}$,

$\varphi, \psi \in \textsc{basic}{\cdot}\textsc{dox} \Rightarrow (\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \to \psi), (\varphi \leftrightarrow \psi), \neg\varphi \in \textsc{basic}{\cdot}\textsc{dox}$,

$\varphi \in \textsc{boole} \Rightarrow \mathbf{B}\varphi,\ \mathbf{b}\varphi,\ \mathbf{K}\varphi,\ \mathbf{k}\varphi \in \textsc{basic}{\cdot}\textsc{dox}$.

Full doxastic language

$\textsc{boole} \subseteq \textsc{full}{\cdot}\textsc{dox}$,

$\varphi, \psi \in \textsc{full}{\cdot}\textsc{dox} \Rightarrow$

$(\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \to \psi), (\varphi \leftrightarrow \psi), \neg\varphi, \mathbf{B}\varphi, \mathbf{b}\varphi, \mathbf{K}\varphi, \mathbf{k}\varphi \in \textsc{full}{\cdot}\textsc{dox}$.

Basic revision language

$\textsc{basic}{\cdot}\textsc{dox} \subseteq \textsc{basic}{\cdot}\textsc{rev}$,

$\varphi, \psi \in \textsc{basic}{\cdot}\textsc{rev} \Rightarrow (\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \to \psi), (\varphi \leftrightarrow \psi), \neg\varphi \in \textsc{basic}{\cdot}\textsc{rev}$,

$(\varphi \in \textsc{boole} \ \ \& \ \ \chi \in \textsc{basic}{\cdot}\textsc{rev}) \Rightarrow [*\varphi]\chi,\ \langle *\varphi\rangle\chi \in \textsc{basic}{\cdot}\textsc{rev}$.

Full revision language

$\textsc{full}{\cdot}\textsc{dox} \subseteq \textsc{full}{\cdot}\textsc{rev}$,

$\varphi, \psi \in \textsc{full}{\cdot}\textsc{rev} \Rightarrow (\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \to \psi), (\varphi \leftrightarrow \psi), \neg\varphi \in \textsc{full}{\cdot}\textsc{rev}$,

$(\varphi \in \textsc{full}{\cdot}\textsc{dox} \ \ \& \ \ \chi \in \textsc{full}{\cdot}\textsc{rev}) \Rightarrow [*\varphi]\chi, \langle *\varphi\rangle\chi \in \textsc{full}{\cdot}\textsc{rev}$.

Unlimited revision language

$\textsc{full}{\cdot}\textsc{dox} \subseteq \textsc{unlim}{\cdot}\textsc{rev}$,

$\varphi, \psi \in \textsc{unlim}{\cdot}\textsc{rev} \Rightarrow$

$(\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \to \psi), (\varphi \leftrightarrow \psi), \neg\varphi, \mathbf{B}\varphi, \mathbf{b}\varphi, \mathbf{K}\varphi, \mathbf{k}\varphi \in \textsc{unlim}{\cdot}\textsc{rev}$,

$(\varphi \in \textsc{unlim}{\cdot}\textsc{rev} \ \ \& \ \ \chi \in \textsc{unlim}{\cdot}\textsc{rev}) \Rightarrow [*\varphi]\chi, \langle *\varphi\rangle\chi \in \textsc{unlim}{\cdot}\textsc{rev}$.

Conditional language

$\textsc{boole} \subseteq \textsc{cond}$,

$\varphi, \psi \in \textsc{cond} \Rightarrow$

$(\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \to \psi), (\varphi \leftrightarrow \psi), \neg\varphi, (\varphi \sqsupset \psi), (\varphi > \psi) \in \textsc{cond}$.

Update language

$\textsc{basic}{\cdot}\textsc{dox} \cup \textsc{cond} \subseteq \textsc{update}$,

$\varphi, \psi \in \textsc{update} \Rightarrow (\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \to \psi), (\varphi \leftrightarrow \psi), \neg\varphi \in \textsc{update}$,

$\varphi \in \textsc{cond} \Rightarrow \mathbf{B}\varphi \in \textsc{update}$,

$(\varphi \in \textsc{cond} \ \ \& \ \ \chi \in \textsc{update}) \Rightarrow [\Diamond\varphi]\chi, \langle\Diamond\varphi\rangle\chi \in \textsc{update}$.

We say that a formula is an *agent formula*, relative to BASIC·REV, FULL·REV, UNLIM·REV, or UPDATE, if it can occur both within the scope of a doxastic operator and within the scope of the star operator or rhombus operator, whichever is appropriate.

2.2 Conditional logic

As remarked above, there is a close connection between the theory of belief change and the theory of conditionals. For this reason, we devote an entire section to this topic, which is also, by our lights, a chapter of modal logic.

2.2.1 Topology

Let U be any nonempty set. A *topology* in U is a set T of subsets of U satisfying two conditions: for all $S \subseteq T$, (i) $\bigcup S \in T$, and (ii) if S is finite and nonempty, then $\bigcap S \in T$. A topology always has at least two elements: U is the union of all subsets of U and is therefore a member, and $\varnothing$ is the union of the empty set of subsets of U and is therefore also a member. The structure (U, T) is called a *topological space*, but when it is clear what the intended topology is one usually refers to U itself as the topological space. The elements of T are said to be *open* sets; a *closed* set is one that is the complement of an open set. In general, a set need not be either open or closed, but on the other hand some sets are both; we will use the term *clopen* (adjective or noun) for the latter. Notice that the complement of a clopen set is clopen and that U and $\varnothing$ are clopen in any topology. The (topological) *closure* of a set X, the smallest closed set that includes X, is defined as the intersection of all closed sets that include X.

A *cover* of a set $X \subseteq U$ is a family C of subsets of U such that $X \subseteq \bigcup C$. A cover, every element of which is an open set, is an *open cover*. If C is a cover of X and there is a family $D \subseteq C$ such that $X \subseteq \bigcup D$, then D is a *subcover* of C of X. A topology T is *compact* if every open cover of the whole space has a finite subcover; a logically equivalent condition is that every family of closed subsets of U whose intersection is empty has a finite subfamily whose intersection is empty. A topology T is *totally separated* if, for any pair of distinct elements of U, one is an element of a clopen set of which the other is not. A *Stone topology* is a topology that is compact and totally separated.

A family B of subsets of T is a *base* for the topology if, for every $X \in T$, there is some family $C \subseteq B$ such that $X = \bigcup C$. In other words, B is a base if every open element is the union of some elements of B. It is not difficult to prove that in a Stone topology, the family of clopen sets forms a base.

Let U be a space with a Stone topology. A *sphere system* or, more colloquially, an *onion* (*in* U) is a nonempty family of closed subsets (*spheres*) of U that satisfies two conditions: it is closed under arbitrary nonempty intersection, and it is linearly ordered by set inclusion. An onion is *trivial* if it contains only one sphere and that sphere is the empty set; hence there is a unique trivial onion, namely $\{\varnothing\}$. The *centre* of an onion O is the set $\bigcap O$, and we say that O is centred on $\bigcap O$; thus the trivial onion is centred on the empty set. We say that an onion O *overlaps with* a set X (we assume that X is a subset of U) if $\bigcup O \cap X \neq \varnothing$. The family of spheres of an onion O that intersect with a set X is denoted by $O \bullet X$. If S is a family of sets and X is the smallest element of S, then we may express this by the notation $X \; \mu \; S$ (thus "mu" is a special case of "epsilon"). It is not difficult to prove that if O is an onion that overlaps with a clopen set X, then there is a smallest sphere in O that intersects with X; in symbols, $\bigcup O \cap X \neq \varnothing \Rightarrow \exists Z(Z \; \mu \; (O \bullet X))$; this important condition is called *the limit condition.*

Sphere systems were introduced by David Lewis (who never called them onions) [127]. Ours differ from his in one notable respect: his spheres, but not ours, are closed under unrestricted union. One particular consequence of Lewis's condition is that the empty

set is an element of every sphere system of his, while our onions may, but need not, contain the empty set as an element.

The reader may find it helpful to think of the clopen sets as the agent's propositions of the frame, and closed sets as possible agent's theories (or theory sets, to use a term of Bengt Hansson). The topological setting may surprise some readers, but it provides an elegant way for keeping tabs on the limit condition.

2.2.2 *Semantics*

Let us say a quadruple (U, P, Q, D) is a *Lewis frame* if (i) U is a Stone space, (ii) P is the set of clopen sets, (iii) Q is a quantity (that is, set) of onions in U, (iv) D is a function (the *onion determiner*) assigning to each element u of U an onion Q, and (v) whenever X and Y are clopen subsets of U, then so are

$$\{u \in U : \forall Z(Z \mu (D(u) \bullet X) \Rightarrow Z \cap X \subseteq Y)\}$$

and

$$\{u \in U : \exists Z(Z \mu (D(u) \bullet X) \ \&\ Z \cap X \cap Y \neq \varnothing)\}.$$

We consider a language COND for conditional logic. A *valuation* in a Lewis frame (U, P, Q, D) is a function from the set LETT of propositional letters to P. A *Lewis model* (U, P, Q, D, V) is a Lewis frame (U, P, Q, D) cum valuation V. We define the truth of a formula in a given Lewis model $\mathcal{M} = (U, P, Q, D, V)$ as follows (we suppress reference to $\mathcal{M}$ in the notation). The definition, which proceeds by induction in the usual way, is relative to a point u of U. We use the notation $[\![\varphi]\!]$ for the set called the *truth set* of φ. If $u \in [\![\varphi]\!]$ we say that φ is *true* at u and may write $u \vDash \varphi$.

$$\begin{aligned}
&[\![\varphi]\!] = V(P), \text{ if } P \text{ is a propositional letter},\\
&[\![\varphi \wedge \psi]\!] = [\![\varphi]\!] \cap [\![\psi]\!],\\
&[\![\varphi \vee \psi]\!] = [\![\varphi]\!] \cup [\![\psi]\!],\\
&[\![\neg\varphi]\!] = U - [\![\varphi]\!], \text{ etc.},\\
&[\![\varphi \sqsupset \psi]\!] = \{u \in U : \forall Z(Z \mu D(u) \bullet [\![\varphi]\!] \Rightarrow Z \cap [\![\varphi]\!] \subseteq [\![\psi]\!])\},\\
&[\![\varphi > \psi]\!] = \{u \in U : \exists Z(Z \mu D(u) \bullet [\![\varphi]\!] \ \&\ Z \cap [\![\varphi]\!] \cap [\![\psi]\!] \neq \varnothing)\}.
\end{aligned}$$

(Note that, thanks to the closure rules on P, including (v), $[\![\varphi]\!]$ is a clopen set, for every formula φ.) We say that a formula is *valid* if it is true at all points in all models. A schema is *valid* if all its instances are valid.

2.2.3 *Postulates for David Lewis's VC and VCU*

First, assume as postulates all tautologies and the rule of modus ponens. Then add the rules

(REA) $\varphi \leftrightarrow \varphi' / (\varphi \sqsupset \theta) \leftrightarrow (\varphi' \sqsupset \theta)$,

(REC) $\theta \leftrightarrow \theta' / (\varphi \sqsupset \theta) \leftrightarrow (\varphi \sqsupset \theta')$

and, as axioms, all instances of the following schemata:

(ML1) $(\varphi \sqsupset (\psi \wedge \theta)) \leftrightarrow ((\varphi \sqsupset \psi) \wedge (\varphi \sqsupset \theta))$,

(ML2) $\varphi \sqsupset \top$,

(DF>) $(\varphi > \psi) \leftrightarrow \neg(\varphi \sqsupset \neg\psi)$,

(CL1) $\varphi \sqsupset \varphi$,

(CL2) $(\varphi > \psi) \rightarrow (\psi > \top)$,

(CL3) $\varphi \rightarrow (\psi \rightarrow (\varphi \sqsupset \psi))$,

(CL4) $\varphi \rightarrow ((\varphi \sqsupset \psi) \rightarrow \psi)$,

(CL5) $((\varphi \wedge \psi) \sqsupset \theta) \rightarrow (\varphi \sqsupset (\psi \rightarrow \theta))$,

(CL6) $(\varphi > \psi) \rightarrow ((\varphi \sqsupset (\psi \rightarrow \theta)) \rightarrow ((\varphi \wedge \psi) \sqsupset \theta))$.

Brian Chellas suggested a different notation which highlights the connection with modal logic: writing $[\varphi]\psi$ for $\varphi \sqsupset \psi$ and $\langle\varphi\rangle\psi$ for $\varphi > \psi$. If we rewrite the preceding conditions in Chellas's notation, we get the following result:

(REA′) $\varphi \leftrightarrow \varphi'/[\varphi]\theta \leftrightarrow [\varphi']\theta$,

(REC′) $\theta \leftrightarrow \theta'/[\varphi]\theta \leftrightarrow [\varphi]\theta'$

and, as axioms, all instances of the following schemata:

(ML1′) $[\varphi](\psi \wedge \theta) \leftrightarrow ([\varphi]\psi) \wedge [\varphi]\theta)$,

(ML2′) $[\varphi]\top$,

(DF⟨...⟩) $\langle\varphi\rangle\psi \leftrightarrow \neg[\varphi]\neg\psi)$,

(CL1′) $[\varphi]\varphi$,

(CL2′) $\langle\varphi\rangle\psi \rightarrow \langle\psi\rangle\top$,

(CL3′) $\varphi \rightarrow (\psi \rightarrow [\varphi]\psi)$,

(CL4′) $\varphi \rightarrow ([\varphi]\psi \rightarrow \psi)$,

(CL5′) $[\varphi \wedge \psi]\theta \rightarrow [\varphi](\psi \rightarrow \theta)$,

(CL6′) $\langle\varphi\rangle\psi \rightarrow ([\varphi](\psi \rightarrow \theta) \rightarrow [\varphi \wedge \psi]\theta)$.

The set of theorems of this axiom system coincides with David Lewis's logic VC. To get his VCU, add the schemata:

(4) $\Box\varphi \rightarrow \Box\Box\varphi$

(5) $\neg\Box\varphi \rightarrow \Box\neg\Box\varphi$,

where $\Box\varphi$ is an abbreviation of $\neg\varphi \sqsupset \bot$ (or $[\neg\varphi]\bot$, in Chellas's notation). (Note that the schema $\Box\varphi \rightarrow \varphi$ is derivable in VC).

We say after Lewis that a frame (U, T, Q, D) is *centred* if the onion $D(u)$ is centred on $\{u\}$ for each $u \in U$, and we say that it is *uniform* if $\bigcup D(u) = \bigcup D(v)$, for all $u, v \in U$.

THEOREM 3 ([127]). *A formula of conditional logic is derivable in VC [alternatively: in VCU] if and only if it is valid in all centred [alternatively: uniform centred] Lewis frames.*

There are of course many more completeness results of this kind.

2.3 *Update and the logic of conditionals*

2.3.1 *Semantics*

Consider an update language UPDATE. Let $\mathcal{M} = (U, P, Q, D, V)$ be a given Lewis model. The definition of truth in $\mathcal{M}$ (we will suppress reference to $\mathcal{M}$ in the notation) will be with respect to a *situation*, defined as an ordered pair (B, u) where B is a subset of U and x is a point in U. (Intuitively, B represents the current beliefs of the agent, while x represents the point that is currently actual.) The definition proceeds in two steps. First, we define the *truth set* $[\![\theta]\!]$ of formulæ $\theta \in$ COND in the usual way; thus $[\![\theta]\!]$ is a subset of U. Second, we define *truth in a situation* (B, x) of formulæ φ in UPDATE:

$$\begin{array}{l}
(B,x) \vDash \varphi \text{ iff } x \in [\![\varphi]\!], \text{ if } \varphi \in \text{ COND},\\
(B,x) \vDash \varphi \wedge \psi \text{ iff } (B,x) \vDash \varphi \text{ and } (B,x) \vDash \psi\\
(B,x) \vDash \varphi \vee \psi \text{ iff } (B,x) \vDash \varphi \text{ or } (B,x) \vDash \psi,\\
(B,x) \vDash \neg\varphi \text{ iff } (B,x) \nvDash \varphi,\\
\qquad \text{etc.},\\
(B,x) \vDash \mathbf{B}\varphi \text{ iff } B \subseteq [\![\varphi]\!],\\
(B,x) \vDash \mathbf{K}\varphi \text{ iff } \bigcup\{\bigcup D(u) : u \in B\} \subseteq [\![\varphi]\!],\\
(B,x) \vDash [\Diamond\varphi]\chi \text{ iff } (B',x) \vDash \chi, \text{ where}\\
\qquad B' = \bigcup\{Z \cap [\![\varphi]\!] : \exists u(u \in B \;\&\; Z \;\mu\; (D(u) \bullet [\![\varphi]\!]))\}.
\end{array}$$

We say that a formula is *valid* if it is true in all situations in all models. A schema is *valid* if all its instances are valid.

2.3.2 *Postulates*

We build an axiom system in stages, one block at the time. First block: all postulates (axioms and rules) of Lewis's VC. Second block: normal modal logic for all modal and dynamic operators. Third block:

($\Diamond$0) $\quad \theta \leftrightarrow [\Diamond\varphi]\theta$, if $\theta \in$ COND,

($\Diamond$1) $\quad \langle\Diamond\varphi\rangle\chi \leftrightarrow [\Diamond\varphi]\chi$,

($\Diamond$RT) $\quad \mathbf{B}(\varphi \sqsupset \psi) \leftrightarrow [\Diamond\varphi]\mathbf{B}\psi$

($\Diamond$K) $\quad \mathbf{K}\varphi \leftrightarrow \mathbf{B}\Box\varphi$

($\Diamond$RC) $\quad$ if $\varphi \leftrightarrow \psi$ is derivable, then so is $[\Diamond\varphi]\chi \leftrightarrow [\Diamond\psi]\chi$, for every χ, for all formulæ $\varphi, \psi \in$ COND and $\chi \in$ UPDATE.

(Here, as above, $\Box\varphi$ abbreviates $\neg\varphi \sqsupset \bot$.) The Ramsey condition is essentially a condition of operator shift where both operators and positions change; this fact is especially striking if ($\Diamond$RT) is rewritten in Chellas's notation:

($\Diamond$RT$'$) $\quad \mathbf{B}[\varphi]\psi \leftrightarrow [\Diamond\varphi]\mathbf{B}\psi$.

Call this system U. (Warning: Lewis's U (for "uniform") must not be be confused with our U (for "update").) Let U45 be the system obtained by adding the schemata (4) and (5) mentioned above.

THEOREM 4. *A formula of the update language is derivable in U [alternatively: in U45] if and only if it is valid in all centred [alternatively: uniform centred] Lewis frames.*

Our axiomatisation, in which ($\Diamond$RT) is the only postulate that is really novel, has thus issued in yet another confirmation of the observation of several authors about the close connection between conditional logic and update logic. It is worth remarking that the class of uniform centred frames, which in our object language determines UT45, also, in another object language which does not include the operator **B**, determines the logic Gösta Grahne calls VCU^2, a logic based directly on Lewis's VCU (see [115]).

2.4 Revision and basic DDL

2.4.1 Semantics

In this section we assume a language BASIC-REV for basic revision logic. Two intuitions underlying our presentation are that belief change consists in the transition from belief state to belief state, and that belief states can be modelled by sphere systems (onions). Let us define a *basic revision frame* as a structure (U, P, Q, R) where U is a Stone space, P is the set of clopen sets, Q is a quantity of onions, and R is a function assigning to each clopen set X a binary relation RX over Q. The intuition is this: if the agent is in belief state O, then after accepting (the information carried by) a proposition X, his new belief state is a belief state O' such that $(O, O') \in RX$.

Valuations and models are defined as usual. We define the *truth* of a formula in a given model $\mathcal{M} = (U, P, Q, R, V)$ as follows (as usual, we suppress reference to $\mathcal{M}$ in the notation). The definition, which proceeds by induction, is relative to a *situation*, which is an ordered pair (O, x), where O is an onion and x a point of U. (Intuitively, O is the current belief state of the agent, and x is the current state of the world.) We use the notation $[\![\varphi]\!]$ for the truth set of φ if φ is a Boolean formula.

$$
\begin{aligned}
&(O, x) \models P \text{ iff } x \in V(P), \text{ if } P \text{ is a propositional letter},\\
&(O, x) \models \varphi \wedge \psi \text{ iff } (O, x) \models \varphi \text{ and } (O, x) \models \psi,\\
&(O, x) \models \varphi \vee \psi \text{ iff } (O, x) \models \varphi \text{ or } (O, x) \models \psi,\\
&(O, x) \models \neg\varphi \text{ iff } (O, x) \not\models \varphi, \text{ etc.},\\
&(O, x) \models \mathbf{B}\varphi \text{ iff } \bigcap O \subseteq [\![\varphi]\!],\\
&(O, x) \models \mathbf{K}\varphi \text{ iff } \bigcup O \subseteq [\![\varphi]\!],\\
&(O, x) \models [*\varphi]\chi \text{ iff } \forall O'((O, O') \in R[\![\varphi]\!] \Rightarrow (O', x) \models \chi).
\end{aligned}
$$

Notice that the truth-conditions for the dynamic operators make sense as long as the star operator applies only to Boolean formulæ. *Validity* in a frame [in a class of frames] is then defined as truth relative to all situations in all models on the frame [in all frames].

To try to capture the ideas behind the historical AGM, further conditions are in order. One is that the belief set of a new belief state resulting from some new piece of information equal the overlap between the old onion and the clopen set representing that information:

(i) $(O, O') \in RX$ only if O overlaps with X and $\bigcap O' = Z \cap X$, where $Z \ \mu\ O \bullet X$.

Two other conditions are that every relation RX be serial and functional:

(ii) $\exists O' \in Q(O, O') \in RX$,

(iii) $(O, O') \in RX$ & $(O, O'') \in RX \Rightarrow O' = O''$.

Yet another condition to be considered is that what AGM called the "background" theory (in our jargon, the agent's doxastic commitments) not change when beliefs are revised:

(iv) $O, O' \in Q \Rightarrow \bigcup O = \bigcup O'$.

2.4.2 Translations of the AGM postulates

A direct translation of the AGM postulates formulated in Section 2.1.2 gives the following result:

(∗2) $[*\varphi]\mathbf{B}\varphi$,

(∗3) $[*\varphi]\mathbf{B}\chi \rightarrow \mathbf{B}(\varphi \rightarrow \chi)$,

(∗4) $\mathbf{b}\varphi \rightarrow (\mathbf{B}\chi \rightarrow [*\varphi]\mathbf{B}\chi)$,

(∗5) $\mathbf{k}\varphi \rightarrow \langle *\varphi\rangle\mathbf{b}\top$,

(∗6) $\mathbf{K}(\varphi \leftrightarrow \psi) \rightarrow ([*\varphi]\mathbf{B}\chi \leftrightarrow [*\psi]\mathbf{B}\chi)$,

(∗7) $[*(\varphi \wedge \psi)]\mathbf{B}\chi \rightarrow [*\varphi]\mathbf{B}(\psi \rightarrow \chi)$,

(∗8) $\langle *\varphi\rangle\mathbf{b}\psi \rightarrow ([*\varphi]\mathbf{B}(\psi \rightarrow \chi) \rightarrow [(\varphi \wedge \psi)]\mathbf{B}\chi$.

All instances of these schemata are valid.

2.4.3 Postulates for the basic-DDL version of AGM

We build an axiom system in stages, one block at the time. First block: tautologies and modus ponens. Second block: normal modal logic for **B** and **K** and $[*\varphi]$. Third block: the postulates (∗2)–(∗8) mentioned in the preceding section plus the following additional postulates:

(∗0) $\chi \leftrightarrow [*\varphi]\chi$, if χ is Boolean,

(∗1) $\langle *\varphi\rangle\chi \leftrightarrow [*\varphi]\chi$,

(∗KB) $\mathbf{K}\varphi \rightarrow \mathbf{B}\varphi$,

(∗K) $\mathbf{K}\chi \leftrightarrow [*\varphi]\mathbf{K}\chi$.

(RC) if $\varphi \leftrightarrow \psi$ is derivable, then so is $[*\varphi]\chi \leftrightarrow [*\psi]\chi$, for every χ.

THEOREM 5 ([167]). *A formula of basic revision language is provable in the given axiom system if and only if it is valid in all basic revision frames.*

2.4.4 Comparison with KM

The two paradigms AGM and KM seem rather different. It is interesting, therefore, that the differences between the corresponding logics is not greater than it is. Of the AGM postulates, all are valid according to KM as well, except for the most controversial ones: $(*4)$ and $(*8)$. But even in these cases the two paradigms come close:

DDL version of KM	DDL version of AGM
$\mathbf{B}\varphi \to (\mathbf{B}\chi \to [*\varphi]\mathbf{B}\chi))$	$\mathbf{b}\varphi \to (\mathbf{B}\chi \to [*\varphi]\mathbf{B}\chi)$
$\langle *\varphi\rangle\mathbf{B}\psi \to$ $([*\varphi]\mathbf{B}(\psi \to \chi) \to [*(\varphi \wedge \psi)]\mathbf{B}\chi)$	$\langle *\varphi\rangle\mathbf{b}\psi \to$ $([*\varphi]\mathbf{B}(\psi \to \chi) \to [*(\varphi \wedge \psi)]\mathbf{B}\chi)$

Among schemata valid in KM but not in AGM are $\mathbf{B}\bot \to [*\varphi]\mathbf{B}\bot$ and $\mathbf{B}\bot \to \mathbf{K}\bot$.

One difference between revision and update, remarked upon by Katsuno and Mendelzon, is what they call the "global" behaviour of revision versus the "local" behaviour of update. What they have in mind can be explicated within our framework as follows. In AGM, the belief states of an agent are represented by onions; the belief set of an agent is not enough to determine the entire onion. By contrast, in KM the belief set is enough to determine the belief state of the agent. The reason for this is of course the centred onions assigned to each point ("possible world") in the universe of a frame. In the latter case, the beliefs of the agent come in two steps: beliefs about the world (represented by centred onions), and beliefs about which possible world is the actual one (represented by a belief set). In AGM, belief change is a progression from onion to onion. In KM, belief change is from belief set to belief set, but against the background of an underlying, constant web of beliefs about how the world can change.

2.5 Revision and full or unlimited DDL

2.5.1 Semantics

Basic DDL tries to explicate AGM as originally formulated. This is why in basic DDL the agent's beliefs are all about the world, and the agent's beliefs are not part of the world. As we saw above, the language of AGM and the language of DDL are intertranslatable. Nevertheless, there is one sense in which DDL offers more flexibility: where AGM has $\chi \in T$, DDL offers $\mathbf{B}\chi$, but AGM has no counterpart to $\mathbf{BB}\chi$ — $(\chi \in T) \in T$ is not a well-formed expression. There is no reason why one could not extend the language of AGM to include the **B**-operator, but no-one seems to have done so. And rather than doing so, it seems easier to study the resulting theory in a DDL context.

Therefore, let us move to the language FULL·REV of full revision, in which the agent formulæ are the formulæ of the full doxastic language FULL·DOX. Define a *full revision frame* as a structure (U, P, Q, R, D) *where* (i) U is a Stone space, (ii) P is the set of clopen sets, (iii) Q is a quantity of onions (not necessarily centred), (iv) R is a function assigning to each clopen set X a binary relation RX over Q, (v) D is a function from $U$to Q and finally (vi) whenever X is a clopen subset of U, then both $\{u \in U : \bigcap D(u) \subseteq X\}$ and $\{u \in U : \bigcup D(u) \subseteq X\}$ are clopen. *Truth* at a point u in a model (U, P, Q, R, D, V) is defined along usual lines:

$$
\begin{aligned}
&u \vDash P \text{ iff } u \in V(P), \text{ if } P \text{ is a propositional letter,}\\
&u \vDash \varphi \wedge \psi \text{ iff } u \vDash \varphi \text{ and } u \vDash \psi\\
&u \vDash \varphi \vee \psi \text{ iff } u \vDash \varphi \text{ or } u \vDash \psi,\\
&u \vDash \neg\varphi \text{ iff } u \nvDash \varphi, \text{ etc.,}\\
&u \vDash \mathbf{B}\varphi \text{ iff } \bigcap O \subseteq [\![\varphi]\!], \text{ where } O = D(u),\\
&u \vDash \mathbf{K}\varphi \text{ iff } \bigcup O \subseteq [\![\varphi]\!], \text{ where } O = D(u),\\
&u \vDash [*\varphi]\chi \text{ iff } \forall v((u,v) \in R[\![\varphi]\!] \Rightarrow v \vDash \chi).
\end{aligned}
$$

As usual, a formula is *valid* in a frame if true at all points in all models.

One difference between basic and full DDL is that in the former case the points of a frame represent world states whereas in the latter case they simultaneously represent both world state and belief state. This is why in basic DDL formulæ have to be evaluated at pairs (O,x) where O represents a belief state and x a world state, while in full DDL formulæ are evaluated at points representing total or combined states. In DDL there is thus an ambiguity in the informal term "world state": in a narrow sense, which excludes the agent's beliefs, the points of basic, but not full, DDL represent world states; but in a wide sense, which includes the agent's beliefs, the points of full, but not basic, DDL represent world states. In any case, the intuition in full DDL is this: if the current total state is u, then if the agent accepts (the information carried by) a proposition X, there will be, immediately afterwards, a new current total state v such that $(u, v) \in RX$.

The semantics of unlimited DDL is the same as that of full DDL, with two exceptions: the language whose formulæ are given a meaning is UNLIM·REV, and the definition of an *unlimited revision frame* is obtained from the definition of a full revision frame by adding condition (vii) if X and Y are clopen subsets of U, then both $\{u \in U : \forall v((u,v) \in RX \Rightarrow v \in Y\}$ and $\{u \in U : \exists v((u,v) \in RX \ \&\ v \in Y\}$ are clopen. Intuitively, the points of an unlimited revision frame represents not only the state of the world and the agent's beliefs about the state of the world but also the agent's beliefs about how the world may change.

2.5.2 Redefining revision?

It is interesting that all the old postulates of basic DDL (but now over the formulæ of FULL·REV or UNLIM·REV) are satisfied. However, there are problems. Suppose, for example, that $\mathbf{b}\varphi$ and $\mathbf{B}\neg B\varphi$ are true in a certain situation. Then $[*\varphi]\mathbf{B}\neg\mathbf{B}\varphi$ follows by preservation. By the Success Postulate, we always have $[*\varphi]\mathbf{B}\varphi$. Hence $[*\varphi]\mathbf{B}(\varphi\wedge\neg\mathbf{B}\varphi)$, by modal logic. Or suppose that $\mathbf{b}\varphi$ and $\mathbf{BB}\neg\varphi$ are true in a certain situation. By the same kind of reasoning $[*\varphi]\mathbf{B}(\varphi \wedge \mathbf{B}\neg\varphi)$ follows. Or, even more problematic, note that both $[*(\varphi\wedge\neg\mathbf{B}\varphi)]\mathbf{B}(\varphi\wedge\neg\mathbf{B}\varphi)$ and $[*(\varphi\wedge\mathbf{B}\neg\varphi)]\mathbf{B}(\varphi\wedge\mathbf{B}\neg\varphi)$ are valid. But if it is true, on a certain occasion, that it is raining in Umeå and Sten, who happens to be visiting at Ojmundsbod far from Umeå, does not believe that it is raining in Umeå, or even believes that it is not raining in Umeå, then surely it should be possible for him to accept this information without incurring doxastic inconsistency?

This problem was first noted and left unresolved in van Linder, van der Hoek and Meyer [129]. Two strategies have later been suggested for dealing with it. One is given in Lindström and Rabinowicz [131] in which it is recommended that one perform a certain contraction before revising one's beliefs. Roughly speaking, before accepting new

information, the agent should give up enough currently held beliefs to make sure that new information does not create doxastic inconsistency (assuming the new information is itself consistent). A different though related strategy is proposed in Segerberg [138] where it is recommended that the notion of revision be redefined. Two possibilities are outlined. In both cases, the star operator $*$ is kept but a new revision operator R is introduced. One suggestion is to define

$$[\mathrm{R}\varphi]\chi =_{\mathrm{df}} [*(\varphi \wedge \mathbf{B}\varphi \wedge \ldots \wedge \mathbf{B}^n\varphi)]\chi$$

and require that the logic of $\mathbf{B}$ be of a certain strength, for example, contain at least the schemata $\mathbf{BB}\varphi \to \mathbf{B}\varphi$ and $\mathbf{B}^n\varphi \to \mathbf{B}^{n+1}\varphi$. The other suggestion is to introduce yet a new doxastic operator $\mathbf{C}$ (for "complete" belief) with the semantics

$$u \vDash \mathbf{C}\varphi \text{ iff } \forall n > 0(u \vDash \mathbf{B}^n\varphi),$$

define

$$[\mathrm{R}\varphi]\chi =_{df} [*(\varphi \wedge \mathbf{C}\varphi)]\chi$$

and then require the logic of $\mathbf{B}$ to validate the schema $\mathbf{BB}\varphi \to \mathbf{B}\varphi$. Evidently, the operator $\mathbf{C}$ in effect represents common belief when only one agent is involved. One may note the validity of the following schemata:

$$\begin{aligned}
&\mathbf{C}\varphi \to \mathbf{B}\varphi,\\
&\mathbf{C}\varphi \to \mathbf{CC}\varphi,\\
&\mathbf{BC}\varphi \leftrightarrow \mathbf{CB}\varphi,\\
&(\mathbf{B}\varphi \wedge \mathbf{C}(\varphi \to \mathbf{B}\varphi)) \to \mathbf{C}\varphi.
\end{aligned}$$

In this modelling, which assumes that the doxastic commitments of agents are not open to revision, it is impossible for an agent who values the consistency of his beliefs to accept either the information that it-is-raining-and-he-does-not-believe-that-it-is-raining or the information that it-is-raining-and-he-believes-that-it-is-not-raining. In the terminology of Roy Sorensen [139], $\varphi \wedge \neg\mathbf{B}\varphi$ and $\varphi \wedge \mathbf{B}\neg\varphi$ represent *blindspots*. In general, φ represents a blindspot at u if $\mathbf{k}\varphi \wedge [\mathrm{R}\varphi]\mathbf{B}\bot$ is true at u — if φ is consistent with what the agent knows but revision by φ leads to an absurd belief set.

3 LOGIC OF ACTION AND DEONTIC LOGIC

For natural reasons, deontic logic has been in the hands of deontic logicians from the beginning. As is the case with all modal logicians who do not concentrate on the purely formal aspect of their discipline, they have been acting as philosophers and as logicians at the same time, and so conceptual issues and technical treatment have been intermingled. It is remarkable that, even though deontic logic has been around for a long time, there is as yet not an accepted body of work that extends very far. What is needed to improve the situation in deontic logic, it seems, is to identify and philosophically discuss basic concepts in greater depth than has been done before. Not least must we develop better logics of action. Modal logic should be in a privileged position to inform such work, or so we argue in this essay.

3.1 Logic of action

3.1.1 Logic of action without actions

In the history of the logic of action, there is a line from Anselm in the eleventh century, restarted in modern times by authors like Kanger, Fitch, and A. R. Anderson, and continued by Chellas, which has recently received its most mature expression yet by Belnap, Perloff and Xu. We quickly sketch a version of a theory in this tradition.

We say that a structure (U, A, T, H, E) is a *frame* if the following conditions are met. U is a set of *points* (informally, representing possible (total) states of the world, just as in dynamic logic). A is a finite set of *agents.* T is a linearly ordered set (we refer to the elements of T as *times*), and a *T-history* is a function from T into U. Let H be a certain set of T-histories (from now on, just histories). If $h \in H$ and $t \in T$, then (h, t) is said to be a *moment.* We define two families of equivalence relations over H. First we define two histories h and g as *coinciding up through t*, in symbols, $h \sim_t g$, if $ht' = gt'$, for all t' not later than t (that is, all t' such that $t' \leq t$). For each moment (h, t), define $H_{h,t} = \{g \in H : h \sim_t g\}$. E is a function assigning to each moment (h, t) and agent $a \in A$ a partitioning $E_{h,t,a}$ of $H_{h,t}$. We say that two histories h and g are *action-equivalent for a at t* if h and g are equivalent under $E_{h,t,a}$ (or, equivalently, if h and g are equivalent under $E_{g,t,a}$); we write $h \approx_{t,a} g$ if this is the case. It is clear that both $\sim_t$ and $\approx_{t,a}$ are equivalence relations.

Consider a classical propositional language (for example, BOOLE — see above) to which is added, for each natural number i, a propositional operator D_i with the informal reading "the agent denoted by i brings it about that". A *valuation* in a frame (U, A, T, H, E) is a function assigning to each natural number an element in A and to each propositional letter a set of moments. A *model* on a frame is the frame together with a valuation. The truth-value (*truth* or *falsity)* of a formula at a moment in a model is defined with respect to moments along traditional lines. The formal condition for the action operator D_i is

$$
\begin{aligned}
(h, t) \vDash D_i\varphi \text{ iff } V(i) = a \;\&\; \exists t_0 < t(\forall g(h \approx_{t_0,a} g \Rightarrow (g, t) \vDash \varphi) \;\&\; \\
\exists g(h \sim_{t_0} g \;\&\; (g, t) \nvDash \varphi)).
\end{aligned}
$$

The idea is that, if a is the agent denoted by i, then *a brings it about, with respect to h and t, that φ*, in symbols $D_i\varphi$, iff there exists a time t_0 earlier than t such that two conditions are satisfied, (i) (the positive condition) that φ be true with respect to g and t, where g is *any* history that is action-equivalent with h for a at t_0, and (ii) (the negative condition) that φ be false with respect to g and t, where g is *some* history coinciding with h up through t_0. In other words, speaking somewhat freely, we might say that the positive condition guarantees that φ is true in a certain important respect, while the negative condition shows the necessity of that guarantee.

From a formal point of view (but not philosophically: see [144, p. 197 f.]) the system sketched may be said to present, more or less in the spirit of Chellas [150], the theory of Belnap and Perloff of an operator called by them the "achievement stit" ("stit" for "**s**ees **t**o **i**t **t**hat"). If the negative condition of the truth definition is omitted and we require T to be the set of all (negative and nonnegative) integers, we get a definition which is essentially that of Chellas's do-operator. (In the latter case, the element t_0 mentioned in the truth condition of D_i should be identified as $t - 1$.)

As an operator of agency, the operator D_i differs from many of its competitors in the literature. For example, in the Chellas version, D_i is a normal modal operator; in

particular, $D_i\top$ and $D_i(\varphi \to \psi) \to (D_i\varphi \to D_i\psi)$ and $D_i(\varphi \wedge \psi) \leftrightarrow (D_i\varphi \wedge D_i\psi)$ are all valid (that is, true at all moments in all models); in the Belnap version, none of them is. In particular, even though $(D_i\varphi \wedge D_i\psi) \to D_i(\varphi \wedge \psi)$ is valid in the Belnap version, $D_i(\varphi \wedge \psi) \to (D_i\varphi \wedge D_i\psi)$ is not; only the weaker $D_i(\varphi \wedge \psi) \to (D_i\varphi \vee D_i\psi)$ is valid.

Sometimes, "i is causally responsible for the fact that φ" would seem to be a better translation of $D_i\varphi$ than "i sees to it that φ". For let φ be "the door is closed" and $D_i\varphi$ thus "the agent sees to it that the door is closed" (presumably equivalent to "the agent closes the door"). The validity of $D_i\varphi \to \varphi$ (in both the Chellas version and the Belnap version) implies that (at a certain moment in a certain model) $D_i\varphi$ is true only if φ is; which seems to mean that the door is closed when the agent closes it. But why close a door that is already closed? On a somewhat related point, notice that, with the truth-condition of D_i as defined, there may be a model such that both $(h, t_1) \vDash D_i\varphi$ and $(h, t_2) \vDash D_i\varphi$, where t_1 and t_2 are times such that $t_1 < t_2$ and h is a history such that $(h, t) \vDash \varphi$ for all times t between t_1 and t_2. On the official stit-reading, the agent closes the door at (h, t_1) as well as at (h, t_2), never mind that the door is closed in h throughout the interval $[t_1, t_2]$. The awkwardness disappears on the alternative reading, according to which the agent is causally responsible at (h, t_1) and (h, t_2) for the door being closed. (But other naïve questions remain: Where is the action? When did the door closing take place? Or are such questions not appropriate?)

The theory presented by Belnap and his collaborators is the culmination of a long development in modal logic; it surpasses all earlier efforts by its sophistication, power and comprehensiveness. One reservation one might have is perhaps the one hinted at in the preceding paragraph: it is a logic of action without actions. No author in the Anselm-Kanger-Chellas line up through Belnap — Davidson belongs to a different tradition — has countenanced the existence of actions in his logic: action talk, yes; ontology of actions, no.[24] For those who would like a representation not only of *action language* but of *action* there is therefore a reason to continue the search for a logic of action worth the name (without, of course, any guarantee that such a thing will ever be found).

3.1.2 What dynamic logic can offer

One liberating effect of the introduction of dynamic logic was that it finally permitted modal logicians to talk about actions and events (without necessarily knowing exactly what they were talking about). The novelty was the introduction of a syntactic category of terms and a corresponding semantic category of — well, what? Formally, the meaning of a term is what modal logicians know as an accessibility relation. But since terms were introduced to formalise the action of programs, it was natural for dynamic logicians to think of these accessibility relations — which relate points-before to points-after — as event types or action types. In fact, Vaughan Pratt himself, the originator of the modal logic of programs, as dynamic logic was called in the early days, remarked that he saw his theory as a logic of computer action.

It would seem that if it is reasonable to represent real propositions (propositions about

[24]One author to whom this remark does not apply is J. F. Horty. Horty, who works within the stit-tradition, explicitly refers to choice cells as actions and, in his book [159], actually refers to them as "actions". In correspondence he has made the following comment: "These actions are only action tokens, however — individual concrete actions. There is no such thing as the action type of "opening a window", for example. There are individual, concrete openings of individual windows, but nothing to group them together."

the world) as sets of points, then it cannot be totally unreasonable to represent real events (events in the world) as sets of paths. Such representation has obvious technical advantages. For one thing, the Boolean operations of union and intersection and even complement become well-defined, and so do the operations of relative product and transitive closure. But it also makes it possible more directly to address the question concerning the relationship between events and action, a topic much debated by philosophers. One example from one of the most interesting philosophers of action:

> The notion of a human act is related to the notion of an event, *i.e.*, a change in the world. What is the nature of this relationship? It would not be right, I think, to call acts a kind or species of events. An act *is* not a change in the world. But many acts may quite appropriately be described as the bringing about or *effecting* ('at will') of a change. To act is, in a sense, to *interfere* with 'the course of nature'. ... To every act ... there corresponds a change or an event in the world. [171, 36f., 39]

One wonders why there should be this difference in theoretical status between propositions and events. Given a certain context, propositions may be true of false; but events may occur or not. A proposition perhaps may be made true or made false; but an event may be brought about or avoided. A proposition may be known or be believed or be given a certain normative status by someone competent to do so; but an event can be foreseen or be remembered, be prescribed or proscribed. One may perceive that a proposition is true; but one may also perceive an event. Some authors such as van Benthem have recognised the analogous position of the two categories, propositions and events, and have tried to give them an even-handed treatment. Unfortunately, philosophers have remained unimpressed so far.

3.1.3 Thinking about change

There already exist modal logical modellings containing interpretations of events, not so much in the philosophical as in computer science literature, [149, 163]. In general, we believe that philosophers have much to learn from the theoretical computer scientists, whose assault on conceptual problems is often fresh and undaunted ("never mind what Aristotle said"). But how are we to make philosophical sense of their constructions and avoid *ad hoc*-ishness?

There is an environment, also referred to as *the world*. The world is always in some (total) *state* or other. The states themselves never change, but the state in which the world is (the currently *actual* state) may change from time to time. The way the world changes is influenced, but perhaps not completely determined, by *agents* outside the world. Furthermore, all change is regular: it takes place according to some *change rule*.

Trying to incorporate these ideas into the semantics of traditional modal logic, think of a *system* as a triple (U, A, C), where U represents the world, A is the set of all agents (assumed to be finitely many), and C is a function representing the change rule. Of these three primitives, two are old: the universe U of *points* (representing total world states) and the set A of agents are as above (for simplicity, we shall think of the agents as a set of integers $\{0, 1, \ldots, n-1\}$, where n is nonnegative (if $n = 0$, then A is empty and the system is agentless)). To describe the change function, the one new primitive, is more complicated. The world is always in some total world state or other. Change in the

world (which for simplicity's sake is assumed to be discrete) is completely described by the change function. Any finite change is from a point-before to a point-after, perhaps via a number of intermediate points; an infinite change can be described in the same way except that there is no point-after. Suppose the world is in a state $u \in U$. Suppose the agents, at this point, make individual "contributions" or "inputs" $i_0, \ldots, i_{n-1}$, respectively, and that there are no further contributions from the agents, nor any other kind of interference. The result will be a change in the world (one possible change is of course the null change), which we may represent as a sequence $p = C(u, i_0, \ldots, i_{n-1})$ of points of which u is the first element; we say that p is the *theoretical result* of the input $(i_0, \ldots, i_{n-1})$. Points to be noted. (1) C depends on the currently actual state of the world but not on any other possible state. (2) The nature of the agents' inputs is not specified (in this modelling). But (3) they are assumed to be outside (not a part of) the world. (4) Since the change rule is represented by a given function, the system as described is deterministic. (It would be nondeterministic if the change rule were represented by a function assigning to each $n+1$-tuple $(u, i_0, \ldots, i_{n-1})$, not a path, but a set of paths. Such indeterminacy would be called *ontic*: not due to limited knowledge on our part, but a property of the system itself.) (5) It is possible for an agent to make no proper contribution. When this happens we say, for book-keeping purposes, that his input is the *null input,* and we use the symbol 0 to denote it.

3.1.4 What a modal logic of actions and agency might look like

Suppose we (as outside observers) witness a certain development taking place in a system. What would it be to have a record of it? Any development in the world consists in the succession of one state after the other, therefore to know the sequence of states, in the order they were realised, would be to have a representation of what took place. But knowing the inputs of the agents would yield a fuller understanding. So perhaps we should think of a record as a certain sequence of elements of type $(u, i_0, \ldots, i_{n-1})$. (There are a number of technical details that would need to be addressed in a rigorous exposition. Let us mention one: if $(u, i_0, \ldots, i_{n-1})$ and $(u', i'_0, \ldots, i'_{n-1})$ are consecutive elements of a history, then u' must be the second element of the path $C(u, i_0, \ldots, i_{n-1})$.)

Technically we may think of a record, from now on more often called a *history*, as a function that assigns to each proper subpath q of a certain path p an n-tuple $(i_0, \ldots, i_{n-1})$; we refer to $(i_0, \ldots, i_{n-1})$ as *the agents' input after q*. The path p is called the *trace* of h; in symbols, $tr(h) = p$. Compared to the "thin" histories of classical modal logic, ours are a lot "fatter", but there is an obvious connexion: the trace of a fat history is a thin history.

As for *events*, we think of them as sets of finite paths (a *path* in U is a sequence of points in U). If p is a path in e (that is, if $p \in e$), then we may say that *p realises e*. So if we witness p played out before our eyes, we also witness a realisation of e. Given this terminology, what would it mean to say that the agents bring about or realise a certain event e? This is a question to which at the present time no-one seems to have an answer. One source of difficulty is the complexity of the many-agent case. The effects of the input of one agent can be modified or completely altered if there are inputs of other agents, either at the same time or later. Agents also sometimes change their mind, thus modifying or altering the effects of their own earlier inputs. For these and other reasons it is often difficult, not only in abstract theory but also in real life, to determine agency

and to allocate (causal) responsibility.

Let us examine one particularly simple case. We say that a history h with trace p is *simple* if $p = C(u, i_0, \ldots, i_{n-1})$, where $i_a \neq 0$ for at least some $a < n$, and, furthermore, $h(q) = (0, \ldots, 0)$, for all $q < p$ such that $q \neq \varnothing$. Thus a simple history is one where the trace is the theoretical result of the initial input. In this very special case, it makes sense to make comments such as the following: the action of the agents was allowed to run its course; the agents brought about the path p; the agents' action realised all events e for which there are paths p', p'' and q such that $q \in e$ and $p = p'qp''$; the agents are (causally) responsible for every event realised by their action.

The one-agent case is of special interest. In this case, we treat the other agents, if any, as part of the background. Technically, if a is an agent, then define the *a-reduct of C* as the function C^a, which assigns to each pair (u,i) the set

$$C^a(u, i) = \{p : \exists i_0, \ldots, i_{n-1}(c_a = i \;\&\; p = C(u, i_0, \ldots, i_{n-1}))\}.$$

In general, $C^a(u, i)$ is not a singleton set, so in this case we face a certain indeterminacy. Note, however, that the latter may be said to be *epistemic* in kind, in contrast to the possible ontic indeterminacy discussed above.

So far we have left open the question about the nature of the agents' input. In this particular case, however, it would be tempting to think that the input i of the agent a consists in calling up a program or plan or, as it was termed in Segerberg [164], routine $\mathbf{r}$ such that, if $\mathbf{r}$ is started with the world in the state u, then the paths in $C^a(u, i)$ correspond to possible developments according to $\mathbf{r}$ (computations according to $\mathbf{r}$, if $\mathbf{r}$ is a program). A mathematician might even go as far as identifying the input, the routine and the corresponding reduct: $\mathbf{r} = C^a(u, \mathbf{r})$.

Summarising this discussion: there are three entities that should not be confused: the routine $\mathbf{r}$; the event of running $\mathbf{r}$; the result of running $\mathbf{r}$ (on a particular occasion or in general).

Carrying out the routine $\mathbf{r}$ is, in a sense, what the agent "really does". In the case of individual physical human action — the case that dominates analytical philosophical discussion of action — an agent's routines may be identified with his ways of moving parts of his body:

> If we interpret the idea of a bodily movement generously, a case can be made for saying that all primitive actions are bodily movements. [...] We never do more than move our bodies: the rest is up to nature. [154, 49 and 50]

But of course by our bodily movements we accomplish many other actions. If we introduce a distinction (not honoured by ordinary language) between *doing* and *realising* an action, we might reserve the former locution for what the agent "really does" or "does directly", and the latter for what the agent may accomplish by his action or "does indirectly". In a modal language we might accordingly introduce event operators does_i, done_i and $\mathsf{realises}_i$, $\mathsf{realised}_i$ and contemplate appropriate meaning-conditions for them. There are no direct counterparts in natural language to these operators, but we have something like the following in mind. Let e be the event that is the interpretation of α

and a the agent assigned to i:

> $\mathsf{does}_i\alpha$: a is just about to do e,
> $\mathsf{done}_i\alpha$: a has just finished doing e,
> $\mathsf{realises}_i\alpha$: because of a's action, e is just about to be realised,
> $\mathsf{realised}_i\alpha$: because of a's action, e has just been realised.

It is a challenge as yet unmet to give fruitful conditions for the latter two operators, which seems like the more important pair. The challenge will not be met in this paper either, but it may be instructive to see what the difficulty is.

Let us say that a history is *well-behaved* if it is the sequence of simple histories. Furthermore, let us say that (h, g) is an *articulated history* if the last element of the trace of h is also the first element of the trace of g; call the latter element the *virtual present* of (h, g). Note that it is natural to think of h as the past history up to the virtual present moment of (h, g) and g as a possible future history from there on. The following semi-formal meaning-conditions summarise the remarks above concerning the one-agent case. Assume that (h, g) is an articulated history and that both h and g are well-behaved. We assume that a is the agent assigned to i, that α is an event term and that the interpretation of an event term α is an event $[\![\alpha]\!]$:

$(h, g) \vDash \mathsf{does}_i\alpha$ iff g begins with an initial simple subhistory g' such that $C^a(u, i) = [\![\alpha]\!]$ and $tr(g') \in [\![\alpha]\!]$, where $i \neq \varnothing$ is a's contribution after h,

$(h, g) \vDash \mathsf{done}_i\alpha$ iff h ends with an terminal simple subhistory h' such that $C^a(v, i) = [\![\alpha]\!]$ and $tr(h') \in [\![\alpha]\!]$, where $h = h''h'$, for some history h'', and v is the first element of $tr(h')$, and $i \neq \varnothing$ is a's contribution after h''.

For the other pair one might try conditions such as these:

$(h, g) \vDash \mathsf{realises}_i\alpha$ iff $\exists g', g'', g_0, g_1, g_2, p(g = g'g''$ & g' is simple & $g' = g_0g_1g_2$ & $p = tr(g_1)$ & $p \in [\![\alpha]\!])$,

$(h, g) \vDash \mathsf{realised}_i\alpha$ iff $\exists h_0, h_1, h_2, g', g'', p(h = h_0h_1h_2$ & $g = g'g''$ & h_1h_2g' is simple & $p = tr(h_2)$ & $p \in [\![\alpha]\!])$

or

$(h, g) \vDash \mathsf{realises}_i\alpha$ iff $\exists h', h'', g_0, g_1, g_2, u, p, i(h = h'h''$ & $g = g_0g_1g_2$ & $h''g_0g_1$ is simple & $p = tr(g_0)$ & $p \in [\![\alpha]\!]$ & u is first in $tr(h'')$ & $tr(h''g') = C^a(u, i)$ & $\forall q(q \in C^a(u, i) \Rightarrow \exists q_0, q_1, q_2(q = q_0q_1q_2$ & $q_1 \in [\![\alpha]\!])))$,

$(h, g) \vDash \mathsf{realised}_i\alpha$ iff $\exists h_0, h_1, h_2, g', g'', u, p, i(h = h_0, h_1, h_2$ & $g = g'g''$ & h_1h_2g' is simple & $p = tr(h_2)$ & $p \in [\![\alpha]\!]$ & u is first in $tr(h_1)$ & $tr(h_1h_2g) = C^a(u, i)$ & $\forall q(q \in C^a(u, i) \Rightarrow \exists q_0, q_1, q_2(q = q_0q_1q_2$ & $q_1 \in [\![\alpha]\!])))$.

Call the former pair of conditions the weak definition and the latter pair the strong definition of action realisation. The weak definition is probably too wide, and the strong definition probably too narrow, for either properly to reflect an intuitive understanding of an action or event being realised. Another complication is that in daily life there is a tendency to consider that an agent has realised an action (whether he intended it or not) if and only if we hold him causally responsible for it; that is, if we can attribute agency to him. Insofar as the attribution of agency is normative in this sense, it is beyond our simple modelling.

3.2 Deontic logic

3.2.1 Background

Deontic logic is the formal study of normative concepts. Here we shall concentrate on the concepts 'obligatory' and 'ought'. There are many 'oughts', and it is well to keep them apart. In particular, we wish to call attention to three distinctions. One that is particularly relevant to this paper is that between 'ought-to-be' (*Seinsollen*) and 'ought-to-do' (*Tunsollen*): 'oughts' that apply to the state of the world, and 'oughts' that apply to actions. That they really are different notions that require different logics was argued particularly forcefully by Castañeda. The distinction itself is older and often associated with Meinong, who seems to have held that, even though they are different concepts, *Tunsollen* is in the final analysis logically reducible to *Seinsollen*. This view was endorsed by Chisholm who gave it a more precise formulation: in his view — dubbed the Meinong/Chisholm thesis by Horty —

(1) it ought to be that i brings it about that φ,

(2) i ought to bring it about that φ,

are logically equivalent [152, 159]. Actually, this thesis involves also another important distinction: that between the *personal* and the *impersonal*. This is seen more clearly if (2) is rephrased as "it is obligatory for i to bring it about that φ" or even "i has an obligation to bring it about that φ". So in a discussion of the Meinong/Chisholm thesis there are actually two distinctions to bear in mind.

Yet another important distinction is that between *standing* versus *one-time* notions of deontic concepts. A standing obligation (permission, prohibition) has a certain scope and covers everything in that scope, while a one-time obligation (permission, prohibition) concerns one particular item (event, occasion, alternative, possibility, or what not). The two kinds of concepts differ in respect to performance of the actions involved. For example, while a one-time obligation is discharged when one performs the particular action it concerns, a standing obligation can be violated but never, within it scope, completely discharged.

To bring analytical order to this field, von Wright 1963 introduced deontic logic. Relatively quickly Standard Deontic Logic emerged, simply classical logic with an extra propositional operator $\mathbf{O}$ with the informal reading of $\mathbf{O}\varphi$ as "it is obligatory that φ" or "it ought to be the case that φ" and satisfying certain extra postulates: all instances of the schemata $\mathbf{O}(\varphi \wedge \psi) \leftrightarrow (\mathbf{O}\varphi \wedge \mathbf{O}\psi)$ and $\mathbf{O}\varphi \rightarrow \neg\mathbf{O}\neg\varphi$ are axioms, and the rule of replacement of provable equivalents holds. Some authors would also include further axiom schemata, for example, $\mathbf{O}\top$ (making the system normal), $\mathbf{O}\varphi \rightarrow \mathbf{O}\mathbf{O}\varphi, \mathbf{O}\mathbf{O}\varphi \rightarrow \mathbf{O}\varphi$ and $\mathbf{O}(\mathbf{O}\varphi \rightarrow \varphi)$. (For definitions and criticisms of SDL, see [156, 153].)

Needless to say, this simple theory — which was a great step forward at the time — was unable to deal with the barrage of counterexamples and conundrums posed by moral philosophers. Some have concluded that the dream of an adequate formal deontic logic is a chimæra, other have looked for ways in which to increase the expressiveness of that very primitive object language. In particular, some authors, including von Wright himself, decided that deontic logic needs a logic of action as a base.

3.2.2 Deontic logic within dynamic logic?

Someone who accepts dynamic logic as a logic of action could reasonably try the simple device of directly adding deontic operators. The latter would of course apply to terms, not formulæ. For example, let us write $\mathsf{ob}_a\alpha, \mathsf{pm}_a\alpha$, and $\mathsf{fb}_a\alpha$ for "α is obligatory for a", "α is permitted for a" and "α is forbidden for a", respectively. In order to define these notions, one might resort to a well-known device going back to Stig Kanger and Alan Ross Anderson, independently of one another, of introducing a constant OK (for approval or absence of a sanction) or a constant S (for disapproval or presence of a sanction); the two approaches are equivalent on the assumption that the formula $\mathsf{S} \leftrightarrow \neg\mathsf{OK}$ is valid. [161, 141, 142] or permission and prohibition this would seem to provide a start, at least initially. In fact, two possibilities come to mind:

(1) $\mathsf{pm}_a\alpha \leftrightarrow [\alpha]\mathsf{OK}$ and $\mathsf{fb}_a\alpha \leftrightarrow \neg[\alpha]\mathsf{OK}$,

(2) $\mathsf{pm}_a\alpha \leftrightarrow \langle\alpha\rangle\mathsf{OK}$ and $\mathsf{fb}_a\alpha \leftrightarrow \neg\langle\alpha\rangle\mathsf{OK}$.

Both have a certain plausibility. Alternative (1) is in the spirit of so-called free-choice permission: permission implies that any outcome of doing the permitted will meet with approval. Alternative (2) is more insidious: if the agent has permission to do something there may nevertheless be outcomes of exercising the permission that will incur the sanction. It seems we do have concepts of permission with these features, say, strong permission and weak permission. There are analogous remarks about prohibition; in either case, the formula $\mathsf{fb}_a\alpha \leftrightarrow \neg\mathsf{pm}_a\alpha$, is valid. So far, so good. But for obligation there is a problem: how to express it? In Standard Deontic Logic, φ is obligatory if and only if the negation of φ is not forbidden. So perhaps one might try $\mathsf{ob}_a\alpha \leftrightarrow \neg\mathsf{pm}_a(-\alpha)$, where pm is one of the alternative operators above and $-\alpha$ is "the complement of α". (See [162] for an effort of this kind; cf. [146].) However, there are difficulties with this approach, which seem hard to overcome. The main difficulty is perhaps that, although the notion of the complement of α can be given a precise meaning in the formal semantics, it does not agree well with intuitive notions. If events are binary relations in a set U, then the complement $U \times U - e$ of an event e is of course again a binary relation. But in general the complement may not be recognised as an intuitively well-defined event corresponding to that set-theoretical entity. It is also worth noticing that sanctions and absence of sanctions may apply not to points but to paths: not so much to *what* is done as *how* it is done. It may be all right to drive from one place to another, but if you do so by going in the wrong direction on a one-way street you may find yourself in trouble. Again, one would wish for a more general analysis.

3.2.3 Norms, norm systems and norm functions

There are norms of different kinds. Every time a mode of behaviour is prescribed or proscribed, approved or disapproved, a norm or a norm system is created. Not only do we have moral and legal codes of varying complexity, but in general all standards of behaviour set norms. 'Etiquette', 'decorum', '*savoir-faire*', '*comme-il-faut*' and 'tasteful' exemplify concepts that are meaningful only in relation to some norm system. The norm systems we meet in daily life are usually neither exact nor complete. For any complex norm system, we need experts, pundits, arbiters, judges, connoisseurs or some such authority to implement it. The ten commandments and the Golden Rule form the

basis of a (religiously founded) morality, but we need theologians to explain what they mean, and ministers to tell us how to apply them. The law attempts to give rules for assessing any possible situation that may come up, but lawyers often disagree about what the law says in a particular case; in many countries, even the Supreme Court decides its issues by vote.

What would it be to have a complete norm system? Consider a given, maximal history. If the norm system, call it the Norm, is complete and we (the analysts) have a full understanding of it, then it should be possible for us, at least in principle, to examine the history, from beginning to end, and see whether at any stage there has been a violation of the Norm. If there has, then paint the history red. Otherwise, ask if there is some over all respect in which the history fails to comply with the Norm. If there is, then paint the history yellow. Finally, if after all this the history is not painted either red or yellow, then paint it green. The set of green histories could be called "legal" if the Norm is legal, "moral" if the Norm is moral, "politically correct" if the Norm is political correctness, and so on. Here, to use a neutral, expression, we shall call the green histories *normal*. At this point, we shall not make a distinction between yellow and red but simply call all histories of that colour *non-normal*.

Strictly speaking, it is not enough that a norm system can partition the set of maximal histories into normal and non-normal; for any past history, the set of future histories must be similarly partitioned. For in general — unless the Norm is totally unforgiving or recognises the possibility of so-called tragic dilemmas or, at the other extreme, is totally tolerant or permissive — any past will admit of possible futures that are red or yellow or green in the sense just described.

In order formally to represent a norm system in this sense — there could of course be several, but we shall be dealing with only one — we now introduce the concept of a norm function. Consider the model theory outlined in the section on the logic of action. Let U and T be as in subsection 3.1.1. A (*total*) *T-history* is a function from T to U; a partial function from T to U is *a partial T-history*. A *past* is a partial history h such that $hg \in H$, for some history g. By the same token, a *future* is a history g such that $hg \in H$ for from some history h. If h is a past, we write $cont(h)$ and $cont^\circ(h)$ for the set of all *complete continuations* of h in H and the set of all *incomplete continuations* of h in H, respectively; in symbols, $cont(h) = \{g : hg \in H\}$ and $cont^\circ(h) = \{g_0 : \exists g_1(hg_0g_1 \in H)\}$. Now, a *norm function* is a function N from the set of all possible pasts to the set of subsets of all possible futures such that, for every possible past $h, N(h) \subseteq cont(h)$. If $g \in N(h)$ and $g = g'g''$, then we say that g' is a *normal continuation of* h *and that* g is a *complete normal continuation of* h.

We end this subsection by noting a number of modal operators that can be introduced with truth conditions in the modelling of the previous section. First, there are three box operators $[\mathsf{x}]$ and corresponding dual diamond operators $\langle\mathsf{x}\rangle$, where x is $\textsc{h}$ ("historically"), $\textsc{d}$ ("deontically") or $\textsc{f}$ ("future"):

$$(h, g) \vDash [\textsc{h}]\varphi \text{ iff } \forall g'(g' \in cont(h) \Rightarrow (h, g') \vDash \varphi).$$
$$(h, g) \vDash [\textsc{d}]\varphi \text{ iff } \forall g'(g' \in norm(h) \Rightarrow (h, g') \vDash \varphi).$$
$$(h, g) \vDash [\textsc{f}]\varphi \text{ iff } \forall h', g'((hg = h'g' \;\&\; \exists f(f \neq \varnothing \;\&\; h' = hf)) \Rightarrow (h', g') \vDash \varphi).$$

$[\textsc{h}]$ is the operator called "historical necessity" by Chellas and "unavoidability" by Thomason. $[\textsc{d}]$ is a deontic operator that should not be automatically translated as "it is obligatory that"; if a reading other than the literal "for every normal continuation" is insisted

on, we recommend "ideally", but care has to be taken not to read too much into that word. $[\text{F}]$ and $\langle\text{F}\rangle$ are Prior's operators G and F, respectively. We are of course able to help us to all the usual temporal operators, including Kamp's UNTIL:

$$\begin{aligned}(h,g) \vDash (\text{UNTIL}\varphi)\chi \text{ iff either } & \forall g', g''(g = g'g'' \Rightarrow (hg', g'') \nvDash \varphi) \\ \text{or else } & \exists g_1, g_2(g = g_1g_2 \ \&\ (hg_1, g_2)\varphi \ \&\ (1) \ \&\ (2))\end{aligned}$$

where (1) and (2) are the following conditions:

(1) $\forall g', g''((g = g'g'' \ \&\ (hg', g'') \vDash \varphi) \Rightarrow \exists g^*(g' = g_1g^*))$,

(2) $\forall k, k'((g = kk' \ \&\ \exists k^*(g_1 = kk^*)) \Rightarrow (hk, k') \vDash \chi))$.

3.2.4 A fragment of dynamic deontic logic

Deontic logic and doxastic logic are often thought to be formally quite similar. Nevertheless, to develop a dynamic deontic logic (DΔL) as a counterpart to the dynamic doxastic logic (DDL) outlined in the previous section would require much effort. Here we shall be content to offer one example of what $D\Delta L$ might look like by considering how an operator of the personal, one-time, ought-to-do type might be definable in our framework.

In other words, what would a meaning-condition for $\mathsf{ob}_i\alpha$ look like if it is to carry something like the intuitive meaning of "e is obligatory for a" or "it is obligatory for a to see to it that e is done" (where e is an event or action and a is an agent)? A careful explication in natural language might run as follows (*first formulation*): "As long as you haven't done whatever it is that you are obligated to do, you are still supposed to do it (if the obligation has not, for some reason, lapsed), never mind violations of the norm that may have taken place in the past; but when you then do it in a normal way, you thereby discharge that particular obligation." Dressing this vernacular suggestion in semi-technical language may give this result (*second formulation*): an event e is *one-time-obligatory for* agent i, given the past history h, if and only if, *if* at the end of any continuation f_0 of h the event e has not yet been realised by i, *then* (i) e is realised in every normal continuation of hf_0, and (ii) if k_0 is an incomplete normal continuation of hf_0 in which e has been realised, then there is a normal continuation of hf_0k_0 in which e is never again realised.

To make this semi-technical version a notch more formal, counterfactually assume that we possess a definition of action realisation (remember that we were not quite able to work one out in the section on action logic). The semi-formal formulation above may the be replaced by the following formal definition (*third formulation*):

$$\begin{aligned}(h,g) \vDash \quad & \mathsf{ob}_i\alpha \text{ iff } \forall f \in \mathit{cont}(h)\forall f_0, f_1((f = f_0f_1 \ \&\ (hf_0, f_1) \nvDash \mathsf{realised}_i\alpha) \Rightarrow \\ & (\forall k \in \mathit{norm}(hf_0)\exists k_0, k_1(k = k_0k_1 \ \&\ (hf_0k_0, k_1) \vDash \mathsf{realised}_i\alpha \ \&\ \\ & \exists l \in \mathit{norm}(hf_0k_0)\forall l_0, l_1(l = l_0l_1 \Rightarrow (hf_0k_0l_0, l_1) \nvDash \mathsf{realised}_i\alpha))).\end{aligned}$$

The final, fully syntactic version of our definition of personal one-time ought-to-do obligation is the following valid schema (*fourth formulation*):

$$\mathsf{ob}_i\alpha \leftrightarrow [\text{H}](\text{UNTIL}\mathsf{realised}_i\alpha)[\text{D}]\langle\text{F}\rangle(\mathsf{realised}_i\alpha \wedge \langle\text{D}\rangle[\text{F}]\neg\mathsf{realised}_i\alpha).$$

The ob-operator is of course only one of a number of operators explicating a notion of obligatoriness (cf. [169]). Complicated as it is, it nevertheless neglects at least one

important aspect, namely, what may be called the Problem of the Implicit Dead-line: the time within which a one-time obligation should be discharged is often not explicit, but there may still be some time limit that is tacitly understood. (See, for example, [155]).

3.2.5 Normative positions

The modelling presented in the last two sections is limited in numerous ways. Some of them would take much effort to overcome. However, there is one particular shortcoming that is philosophically important and deserves a comment here. Norm systems are usually systematic. If one past history is a (not necessarily normal) continuation of another, then one would expect the normative situation after the former to be in some intimate sense related to the normative situation after the latter: a norm that does not possess a certain minimum of coherence will not be viable. In order to give an example of a possible coherence criterion, let us first augment the modelling of the previous section.

Describing the AGM paradigm above, we introduced sphere systems to model belief states (see subsection 2.2.1). Now we shall use sphere systems to model what we will call the "normative position". Thus we redefine the concept of a *norm function* to be a function N from the set of all possible pasts such that, for every possible past $h, N(h)$ is a sphere system in $cont(h)$ (meaning that $X \subseteq cont(h)$, for every element $X \in N(h)$). It is the sphere system $N(h)$ that we term the *normative position after h*.

We adopt the following technical definitions. If h is a past history, g an incomplete continuation of h, and X is a set of complete continuations of h, then we write X^g for the set of continuations of hg that are final segments of elements of X. Schematically, if h is a past history and $g \in cont^\circ(h)$, then for all $X \subseteq cont(h)$,

$$X^g =_{df} \{f : f \in cont(hg) \; \& \; gf \in X\}.$$

Similarly, if S is a set of subsets of $cont(h)$, then we write S^g for the set of nonempty subsets Y of $cont(hg)$ such that $Y = X^g$, for some $X \in H$. Schematically,

$$S^g = \{X^g : X \in H \; \& \; X^g \neq \varnothing\}.$$

Note that S^g is a sphere system in $cont(hg)$ if S is a sphere system in $cont(h)$.

We are now ready for the definition that is the point of this exercise: we define a norm function as *coherent* if, for all finite continuations g of any past history h,

$$N(hg) = (N(h))^g.$$

Technically, the change from $N(h)$ to $N(hg)$ is related to the notion of irrevocable change discussed in a context of belief revision in [166]. (It is worth recalling that the original concern of Carlos Alchourrón — professor of jurisprudence and one of the fathers of AGM — was, not belief change, but legal change.)

3.2.6 Moral

We have included the non-standard material found in the last few sections — no doubt trying the reader's patience to the limit — in the hope of driving home four theses:

- as first argued by von Wright, deontic logic depends on the logic of action,

- as argued by Castañeda, for logical analysis propositions (formulæ) may not be enough — we also need something like actions (terms),
- the logic of even common normative concepts is more complex than is usually thought,
- modal logic is well equipped to deal with that particular kind of complexity.

ACKNOWLEDGEMENTS

In writing this paper we have benefited from the advice of a select group of colleagues: Joseph Almog, Robert Demolombe, Vincent Hendricks, Jeff Horty. Most of their advice was gratefully followed. The reason for not following all of it — in particular not the most radical suggestion, to go back to square one and write a completely new article — is a familiar one: not enough time and not enough space. We are indebted to them for all their careful, extensive and, on the whole, encouraging comments. Lindström's work was supported by The Bank of Sweden Tercentenary Foundation via the research project "Medvetande, Materialism och Möjlighet" ("Mind, Materialism, and Modality").

BIBLIOGRAPHY

The following bibliography contains not only references in the narrow sense of being explicitly referred to in the main text but also a number of works that it has seemed important to include as a help to those readers who might wish to pursue interests that the authors hope the reading of this essay has stimulated.

References to part 1

[1] J. Almog. Naming without necessity, *The Journal of Philosophy* 83, 210–242, 1986.

[2] A. Anderson. Some new axioms for the logic of sense and denotation: Alternative (0), *Noûs* 14, 217–234, 1980.

[3] A. Anderson. Alonzo Church's contribution to philosophy and intensional logic, *The Bulletin of Symbolic Logic* 4, 129–171, 1998.

[4] R. Ballarin. Validity and necessity, *Journal of Philosophical Logic*, 34, 275–303, 2005.

[5] R. Barcan Marcus. A functional calculus of first order based on strict implication', *The Journal of Symbolic Logic* 11, 1–16, 1946.

[6] R. Barcan Marcus. The deduction theorem in a functional calculus of first order based on strict implication, *The Journal of Symbolic Logic* 11, 115–118, 1956.

[7] R. Barcan Marcus. The identity of individuals in a strict functional calculus of second order, *The Journal of Symbolic Logic* 12, 12–15, 1947.

[8] R. Barcan Marcus. Essentialism in modal logic, *Noûs* 1, 90–96, 1967. Reprinted in Barcan Marcus [9].

[9] R. Barcan Marcus. *Modalities: Philosophical Essays*, Oxford: Oxford University Press, 1993.

[10] J. van Benthem and A. ter Meulen, eds. *Handbook of Logic and Language*, Amsterdam: Elsevier, 1997.

[11] E. W. Beth. Semantic entailment and formal derivability, *Mededelingen van de Koninklijke Nederlandse Akademie van Wetenschappen, Afdeling Letterkunde*, n.s., Vol.18, no. 13. Amsterdam: 1955, 309–42. Reprinted in Hintikka [43, 9–41].

[12] P. Blackburn, M. de Rijke and Y. Venema. *Modal logic*, Cambridge Tracts in Computer Science 53, Cambridge: Cambridge University Press, 2001.

[13] G. Boolos. *The Logic of Provability*, Cambridge: Cambridge University Press, 1993.

[14] J. P. Burgess. Quinus ab Omni Naevo Vindicatus, in Ali Kazmi (ed.) *Meaning and Reference*, *Canadian Journal of Philosophy*, Supplementary Volume 23, 1997.

[15] J. P. Burgess. Which modal logic is the right one?, *Notre Dame Journal of Formal Logic* 40, 81–93, 1999.

[16] R. Carnap. Modalities and quantification, *The Journal of Symbolic Logic* 11, 33–64, 1946.

[17] R. Carnap. *Meaning and Necessity: A Study in Semantics and Modal Logic,* Chicago: University of Chicago Press, 1947. Second edition with supplements, 1956.

[18] A. Church. A formulation of the logic of sense and denotation (abstract), *The Journal of Symbolic Logic* 12, 31, 1946.

[19] A. Church. A formulation of the logic of sense and denotation, in *Structure, Method, and Meaning: Essays in honor of H. M. Sheffer,* New York: Liberal Arts Press, 1951.

[20] A. Church. Outline of a revised formulation of the logic of sense and denotation (Part I), *Noûs* 7, 24–33, 1973.

[21] A. Church. Outline of a revised formulation of the logic of sense and denotation (Part II), *Noûs* 8, 135–156, 1974.

[22] A. Church. Russellian simple type theory, in *Proceedings and Addresses of the American Philosophical Association* 47, 21–33, 1974.

[23] N. Cocchiarella. On the primary and secondary semantics of logical necessity, *Journal of Philosophical Logic* 4, 13–27, 1975.

[24] J. Copeland. The genesis of possible worlds semantics, *Journal of Philosophical Logic* 31, 99–137, 2002.

[25] J. Divers. *Possible Worlds*, London: Routledge. 2002.

[26] K. Fine. Modality *de re*, in Almog *et al. (*eds.*) Themes from Kaplan*, Oxford: Oxford University Press, 197–272, 1986.

[27] K. Fine. Quine on quantifying in, in C. A. Anderson and J. Owens (eds.) *Propositional Attitudes*, Stanford: CSLI, 1–25, 1991.

[28] M. Fitting. *Types, Tableaux, and Gödel's God*, Boston: Kluwer, 2002.

[29] M. Fitting. Intensional logic–beyond first-order, in Hendricks, V., Malinowski, J. (eds.), *50 Years of Studia Logica*, Dordrecht, Kluwer, 87–108, 2003.

[30] M. Fitting and R. Mendelsohn. *First-Order Modal Logic*, Dordrecht: Kluwer, 1998.

[31] D. Føllesdal. *Referential Opacity and Modal Logic*, New York, NY: Routledge, 2004. Published version of doctoral dissertation, Harvard University, 1961.

[32] G. Frege. Über Sinn und Bedeutung, original in *Zeitschrift für Philosophie und philosophische Kritik* 100, 25–50, 1892. English translation in Geach and Black [34].

[33] D. Gallin. *Intensional and Higher-Order Modal Logic*, Amsterdam: North-Holland, 1975.

[34] P. Geach and M. Black, eds. *Translations from the Philosophical Writings of Gottlob Frege*, Oxford: Blackwell 1982.

[35] K. Gödel. Eine Interpretation des intuitionistischen Aussagenkalküls, *Ergebnisse eines mathematischen Kolloquiums*, 4, 39–40, 1933. Reprinted with an English translation ('An interpretation of intuitionistic propositional calculus') in S. Feferman *et al.* (eds.), *Kurt Gödel, Collected Works* Vol. I, Oxford: Oxford University Press, 1986, 300–303.

[36] R. Goldblatt. Mathematical modal logic: a view of its evolution, *Journal of Applied Logic* 1, 309–392, 2003.

[37] A. Hazen. Counterpart-theoretic semantics for modal logic, *The Journal of Philosophy* 76, 319–338, 1979.

[38] J. Hintikka. Form and content in quantification theory, *Acta Philosophica Fennica*, 8, 7–55, 1955.

[39] J. Hintikka. Quantifiers in deontic logic, Helsinki: *Societas Scientarum Fennica, Commentationes Humanarum Literarum* 23, no. 4, 1957.

[40] J. Hintikka. Modality as referential multiplicity, *Ajatus* 20, 49–64, 1957.

[41] J. Hintikka. Modality and quantification, *Theoria* 27, 117–128, 1961.

[42] J. Hintikka. *Knowledge and belief: An Introduction to the Logic of the Two Notions*. Cornell: Cornell University Press, 1962.

[43] J. Hintikka. *The Philosophy of Mathematics*, Oxford Readings in Philosophy, Oxford: Oxford University Press, 1969.

[44] J. Hintikka. *Models for Modalities, Selected Essays*, Dordrecht: D. Reidel, 1969.

[45] J. Hintikka. *The Intentions of Intentionality and Other New Models for Modalities*, Dordrecht: D. Reidel, 1975.

[46] J. Hintikka. Standard vs. nonstandard logic: Higher-order, modal, and first-order logics, in E. Agazzi (ed.) *Modern Logic*, Dordrecht: D. Reidel, 283–296, 1981.

[47] J. Hintikka. Is alethic modal logic possible? In J. Hintikka and M. B. Hintikka: *The Logic of Epistemology and the Epistemology of Logic*. Dordrecht: D. Reidel, 1989.

[48] J. Hintikka and G. Sandu. The fallacies of the new theory of reference, *Synthese* 104, 245–283, 1995.

[49] G. Holmström-Hintikka, S. Lindström, and R. Sliwinski, eds. *Collected Papers of Stig Kanger with Essays on his Life and Work*, Volume I, Synthese Library Vol. 303, Dordrecht: Kluwer, 2001.

[50] G. Holmström-Hintikka, S. Lindström, and R. Sliwinski, eds. *Collected Papers of Stig Kanger with Essays on his Life and Work*, Volume II, Synthese Library Vol. 304, Dordrecht: Kluwer, 2001.

[51] L. Humberstone. Two-dimensional adventures, *Philosophical Studies* 118, 17–65, 2004.

[52] B. Jónsson and A. Tarski. Boolean algebras with operators, *American Journal of Mathematics* 73 (1951), 891–939, vol. 74 (1952), 127–162, 1951.

[53] S. Kanger. *Provability in Logic,* Acta Universitatis Stockholmiensis, *Stockholm Studies in Philosophy* 1, Stockholm: Almqvist & Wiksell, 1957. Reprinted in Holmström-Hintikka *et al.*, [49].

[54] S. Kanger. The morning star paradox, *Theoria* 23, 1–11, 1957. Reprinted in Holmström-Hintikka *et al.*, [49].

[55] S. Kanger. A note on quantification and modalities, *Theoria* 23, 133–134, 1957. Reprinted in Holmström-Hintikka *et al.*, [49].

[56] S. Kanger. On the characterization of modalities, *Theoria* 23, 152–155, 1957. Reprinted in Holmström-Hintikka *et al.*, [49].

[57] S. Kanger. *New foundations for ethical theory, Part 1*, (mimeographed) Stockholm, 1957, reprinted (with minor changes) in R. Hilpinen (ed.): *Deontic logic: introductory and systematic readings*, Dordrecht: D. Reidel 1971, 36–58. Reprinted in Holmström-Hintikka *et al.*, [49].

[58] D. Kaplan. *Foundations of Intensional Logic,* dissertation, University of California, Los Angeles, Ann Arbor: University Microfilms International, 1964.

[59] D. Kaplan. Quantifying in, in D. Davidson and J. Hintikka (eds.), *Words and Objections: Essays on the Work of W. V. Quine,* Dordrecht: D. Reidel, 178–214, 1969.

[60] D. Kaplan. How to Russell a Frege-Church, *The Journal of Philosophy* 72, 716-729, 1975.

[61] D. Kaplan. Opacity, in Hahn and Schilpp (eds.), *The Philosophy of W. V. Quine*, *The Library of Living Philosophers*, Volume XVIII, Open Court, La Salle, Illinois, 1986.

[62] J. Kim. *Supervenience and Mind: Selected Philosphical Essays*, Cambridge and New York: Cambridge University Press, 1993.

[63] J. C. King. Structured Propositions, *The Stanford Encyclopedia of Philosophy (Fall 2001 Edition)*, Edward N. Zalta (ed.), `http://plato.stanford.edu/archives/fall2001/entries/propositions-structured/`

[64] S. Kripke. A completeness theorem in modal logic,*The Journal of Symbolic Logic,* 24, 1–14, 1959.

[65] S. Kripke. Semantical analysis of modal logic (abstract) *The Journal of Symbolic Logic* 24, 323–324, 1959.

[66] S. Kripke. Semantical considerations on modal logic, in *Proceedings of a Colloquium on Modal and Many-Valued Logics,* Helsinki, 23–26 August, 1962. *Acta Philosophica Fennica* 16, 83–94, 1963. Reprinted in Linsky [82].

[67] S. Kripke. Semantical analysis of modal logic: I. Normal Modal Propositional Calculi, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 9, 67–96, 1963.

[68] S. Kripke. Semantical analysis of modal logic: II. Non-normal modal propositional calculi, in J. W. Addison, L. Henkin, and A. Tarski (eds.), *The Theory of Models* (Proceedings of the 1963 International Symposium at Berkeley), Amsterdam: North-Holland, 206–220, 1965.

[69] S. Kripke. Identity and necessity, in M. Munitz (ed.), *Identity and Individuation,* New York: New York University Press, 135–164, 1971.

[70] S. Kripke. Naming and necessity, in D. Davidson and G. Harman (eds.) *Semantics of Natural Language*. Dordrecht: D. Reidel, 253-355, and 763–769, 1972. Reprinted (with a new preface) as Kripke [71].

[71] S. Kripke. *Naming and Necessity*, Cambridge, Mass: Harvard University Press, 1980.

[72] C. I. Lewis. Implication and the algebra of logic, *Mind* 21, 522–531, 1912.

[73] C. I. Lewis. *Survey of Symbolic logic*. Berkeley: University of California Press, 1918.

[74] C. I. Lewis and C. H. Langford. *Symbolic Logic*, New York: The Century Company, 1932.

[75] D. Lewis. Counterpart theory and quantified modal logic, *The Journal of Philosophy* 65, 113–126, 1968.

[76] D. Lewis. General Semantics, *Synthese* 22, 18–67. Reprinted in Lewis [79], 1970.

[77] D. Lewis. *Counterfactuals*, London: Blackwell, 1973.

[78] D. Lewis. Causation, *Journal of Philosophy* 70, 556–67, 1973.

[79] D. Lewis. *Philosophical Papers, Volume I*, Oxford: Oxford University Press, 1983.

[80] D. Lewis. *On the Plurality of Worlds*. London: Blackwell, 1986.

[81] S. Lindström. An exposition and development of Kanger's early semantics for modal logic, in P. W. Humphreys and J. H. Fetzer (eds.) *The New Theory of Reference — Kripke, Marcus, and Its Origins*. Dordrecht: Kluwer, 1998. Reprinted with minor changes in Holmström-Hintikka et al., [49].

[82] L. Linsky. *Reference and Modality*, Oxford: Oxford University Press, 1971.

[83] R. Montague. Logical necessity, physical necessity, ethics and quantifiers, *Inquiry* 4, 259–269, 1960. Reprinted in [84].

[84] R. Montague. *Formal Philosophy: Selected Papers of Richard Montague*. Edited and with an Introduction by Richmond H. Thomason. New Haven: Yale University Press, 1974.

[85] J. Myhill. Problems arising in the formalization of intensional logic, *Logique et Analyse* 1, 78–83, 1958.

[86] S. Neale. On a milestone of empiricism, in Orenstein, A. and P. Kotatko (eds.) *Knowledge, Language, and Logic*, 237–346, 2000.
[87] D. Nolan. *Topics in the Philosophy of Possible Worlds*, London: Routledge, 2002.
[88] C. Parsons. Intensional Logic in Extensional Language, *The Journal of Symbolic Logic* 47, 289–328, 1982.
[89] T. Parsons. Essentialism and quantified modal logic, *The Philosophical Review* 78, 35–52, 1969. Reprinted in Linsky [82].
[90] B. H. Partee and H. L. W. Hendriks. Montague grammar, in van Benthem and ter Meulen [10].
[91] W. V. Quine. Notes on existence and necessity, *Journal of Philosophy* 40, 113–127, 1943.
[92] W. V. Quine. The problem of interpreting modal logic, *The Journal of Symbolic Logic* 12, 43–48, 1947.
[93] W. V. Quine. Reference and modality, in *From a Logical Point of View,* Cambridge, Mass.: Harvard University Press, 1953.
[94] W. V. Quine. Three grades of modal involvement, *Proceedings of the XIth International Congress of Philosophy,* vol. 14, North-Holland, Amsterdam, 65–81, 1953. Reprinted in Quine [95].
[95] W. V. Quine. *The Ways of Paradox and Other Essays,* Cambridge, Mass.: Harvard University Press, 1966.
[96] W. V. Quine. Review of *Identity and Individuation*, M. K. Munitz, ed., New York, 1971. *Journal of Philosophy* 69, 488–97, 1972.
[97] N. Salmon. *Reference and Essence*, Oxford: Basil Blackwell, 1982.
[98] N. Salmon. *Frege's puzzle*, Cambridge, Mass.: MIT Press, 1986.
[99] A. Smullyan. Modality and description, *The Journal of Symbolic Logic* 13, 31–37, 1948.
[100] S. Soames. *Beyond Rigidity: The Unfinished Semantic Agenda of Naming and Necessity*, Oxford: Oxford University Press, 2002.
[101] R. Solovay. Provability interpretations of modal logic, *Israel Journal of Mathematics* 25, 287–304, 1976.
[102] R. Stalnaker. *Context and Content — Essays on Intentionality in Speech and Thought*, Oxford: Oxford University Press, 1999.
[103] R. Stalnaker. *Ways a World Might Be — Metaphysical and Anti-Metaphysical Essays.* Oxford: Oxford University Press, 2003.
[104] Z. G. Szabó. Compositionality, *The Stanford Encyclopedia of Philosophy* (Fall 2004 Edition), Edward N. Zalta (ed.), 2004. `http://plato.stanford.edu/archives/fall2004/entries/compositionality/`
[105] A. Tarski. Der Wahrheitsbegriff in den formalisierten Sprachen, *Studia Philosophica* 1, 261–405, 1936. (English translation: 'The concept of truth in formalized languages', 152–278 in *Logic, Semantics, Metamathematics*, second edition, Hackett Indianapolis, 1983).
[106] A. Tarski. Über den Begriff der logischen Folgerung, *Actes du Congrès International de Philosophie Scientifique*, vol. 7, Paris, 1–11, 1936. (English translation: 'On the concept of logical consequence', 409–420 in *Logic, Semantics, Metamathematics*, second edition, Hackett, Indianapolis, 1983).
[107] G. H. von Wright. *An Essay in Modal Logic*, Amsterdam: North-Holland, 1951.
[108] E. Zalta. Logical and analytic truths that are not necessary, *The Journal of Philosophy*, Vol 85, 57-74, 1988.

References to part 2

[109] C. Alchourrón, P. Gärdenfors and D. Makinson. On the logic of theory change, *The Journal of Symbolic Logic* 50, 510–530, 1985.
[110] H. Arló-Costa. First order extensions of classical systems of modal logic: The role of the Barcan schemas, *Studia Logica* 71, 87–18, 2002.
[111] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*, Cambridge: MIT Press, 1995.
[112] P. Gärdenfors. Imaging and conditionalization, *The Journal of Philosophy* 79, 747–760, 1982.
[113] P. Gärdenfors. *Knowledge in flux: modeling the dynamics of epistemic states.* Cambridge, Mass.: The MIT Press, 1988.
[114] P. Gochet and P. Gribomont. Epistemic Logic, in *Handbook of the History of Logic, Volume VII*, edited by Gabbay, D.M. and Woods, J. Amsterdam: Elsevier Science, 2006.
[115] G. Grahne. Updates and counterfactuals. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, edited by J. A. Allen, R. Fikes & E. Sandewall, 269–276. San Mateo, Calif.: Morgan Kaufmann, 1991.
[116] A. Grove. Two modellings for theory choice, *Journal of Philosophical Logic* 17, 157–170, 1988.

[117] S. O. Hansson. *A Textbook of Belief Dynamics: Theory Change and Data Base Updating*, Dordrecht: Kluwer, 1999.

[118] V. F. Hendricks. *Forcing Epistemology*, forthcoming, Cambridge University Press.

[119] V. F. Hendricks, S. A. Pedersen, and K. F. Jørgensen, eds. *Knowledge Contributors*, Synthese Library Vol. 322. Kluwer Academic Publishers, 2003.

[120] J. Hintikka. *Knowledge and Belief: An Introduction to the Logic of the Two Notions.* Ithaca, NY: Cornell University Press, 1962.

[121] W. van der Hoek, H. von Ditmarsch, and B. Kooi. Concurrent Dynamic Epistemic Logic, in (Hendricks *et al.*, 2003)

[122] H. Katsuno and A. O. Mendelzon. On the difference between updating a knowledge base and revising it. In *Belief revision*, edited by P. Gärdenfors, 183–203. Cambridge: Cambridge University Press, 1992.

[123] A. M. Keller and M. Winslett Wilkins. On the use of an extensional relational model to handle changing incomplete information, *I.E.E.E. transactions on software engineering*, SE-11:7, 620–633, 1985.

[124] H. Levesque. All I know, a study in autoepistemic logic, *Artificial Intelligence* 42, 263–309, 1990.

[125] I. Levi. *The Enterprise of Knowledge*, Cambridge: Cambridge University Press, 1983.

[126] I. Levi. *The Fixation of Belief and its Undoing*, Cambridge: Cambridge University Press, 1991.

[127] D. Lewis. *Counterfactuals.* Oxford: Blackwell, 1973.

[128] D. Lewis. Probabilities of conditionals and conditional probabilities, *Philosophical Review* 85, 297–315, 1976.

[129] B. van Linder, W. van der Hoek, and J.-J. Meyer. Actions that make you change your mind. In *Knowledge and belief in philosophy and artificial intelligence*, edited by A. Laux and H. Wansing, 103–136. Berlin: Akademie-Verlag GmbH, 1995.

[130] S. Lindström and W. Rabinowicz. Epistemic entrenchment with incomparabilities and relational belief revision. In *The logic of theory change*, edited by André Fuhrmann and Michael Morreau, 93–126. *Lecture Notes in Artificial Intelligence*, no. 465. Berlin: Springer-Verlag, 1990.

[131] S. Lindström and W. Rabinowicz. Extending dynamic doxastic logic accommodating iterated beliefs and Ramsey conditionals within DDL. In *For Good Measure: Philosophical Essays Dedicated to Jan Odelstad*, edited by Lars Lindahl, Paul Needham and Rysiek Sliwinski, Uppsala: Uppsala Philosophical Studies 46, 126–153, 1997.

[132] S. Lindström and W. Rabinowicz. DDL Unlimited: Dynamic Doxastic Logic for Introspective Agents', *Erkenntnis* 50, 353–385, 1999.

[133] F. P. Ramsey. *Foundations: Essays in Philosophy, Logic, Mathematics and Economics.* Edited by D. H. Mellor. London and Henley: Routledge and Kegan Paul, 1978.

[134] M. de Rijke. Meeting some neighbours: A dynamic modal logic meets theories of change and knowledge representation, in Van Eijck, J. and Visser, A. (eds.), *Logic and Information Flow*, Cambridge: MIT Press, 1994.

[135] H. Rott. *Change, choice and inference: a study of belief revision and nonmonotonic reasoning.* Oxford Logic Guides Vol. 42. Oxford: Clarendon Press, 2001.

[136] K. Segerberg. Two traditions in the logic of belief: bringing them together. In *Logic, Language and Reasoning: Essays in Honour of Dov Gabbay*, edited by Hans Jürgen Ohlbach and Uwe Reyle, 135–147. Dordrecht: Kluwer, 1999.

[137] K. Segerberg. The basic dynamic doxastic logic of AGM. In *Frontiers in Belief Revision*, edited by Mary-Anne Williams and Hans Rott, 57–84. Dordrecht: Kluwer, 2001.

[138] K. Segerberg. Moore problems in full dynamic doxastic logic. In *Essays in Logic and Ontology: Dedicated to Jerzy Perzanowski*, edited by J. Malinowski and A. Pietruszczak, 11–25. Poznań Studies in the Philosophy of the Sciences and the Humanities, Vol. 50. Amsterdam/Atlanta, GA: Rodopi, 2003.

[139] R. Sorensen. *Blindspots*, Oxford: Oxford University Press, 1988.

[140] R. Stalnaker. A theory of conditionals. In *Studies in Logical Theory*, *American Philosophical Quarterly*, Monograph 2, edited by N. Rescher, 98–112. Oxford: Blackwell, 1968.

References to part 3

[141] A. R. Anderson. The formal analysis of normative systems. Technical report. New Haven, CT: Office of Naval Research, 1962.

[142] A. R. Anderson. Logic, norms, and roles, *Ratio* 4, 36–39, 1962.

[143] L. Åqvist. *Introduction to Deontic Logic and the Theory of Normative Systems.* Napoli: Bibliopolis, 1987.

[144] N. Belnap, M. Perloff and Ming Xu. *Facing the Future: Agents and Choice in Our Indeterministic World.* With contributions by Paul Bartha, Michell Green and John Horty. New York, NY: Oxford University Press, 2001.
[145] J. van Benthem. *Exploring Logical Dynamics*, CSLI Publications, Stanford, Calif, 1996.
[146] J. M. Broersen. *Modal Action Logic for Reasoning about Reactive Systems.* Doctoral dissertation, Vrije Universiteit van Amsterdam, 2003.
[147] H.-N. Castãneda. *Thinking and Doing.* Dordrecht: D. Reidel, 1975.
[148] H.-N. Castãneda. The paradoxes of deontic logic: the simplest solution to all of them in one fell swoop. In *New Studies in Deontic Logic: Norms, Actions and the Foundations of Ethics*, edited by R. Hilpinen, 37–85. Dordrecht: D. Reidel, 1981.
[149] P. R. Cohen and H. J. Levesque. Intention is choice with commitment, *Artificial intelligence*, 42, 213–261, 1990.
[150] B. F. Chellas. *The Logical Form of Imperatives.* Stanford, CA: Perry Lane Press, 1969.
[151] B. F. Chellas. Conditional obligation. In S. Stenlund (ed.), *Logical Theory and Semantic Analysis*, 23–33, Dordrecht: D. Reidel, 1974.
[152] R. Chisholm. The ethics of requirement, *American Philosophical Quarterly* 1, 147–153, 1964.
[153] S. Danielsson. What shall we do with deontic logic?, *Theoria* 66, 97–114, 2000.
[154] D. Davidson. *Essays on Actions and Events.* Oxford: Clarendon Press, 1980.
[155] R. Demolombe, P. Bretier and V. Louis, manuscript, 'Formalisation de l'obligation de faire avec délais'.
[156] J. Forrester. *Being Good and Being Logical: Philosophical Groundwork for a New Deontic Logic.* Armonk, NY and London: M. E. Sharpe, 1996.
[157] R. Hilpinen. Actions in deontic logic. In *Deontic Logic in Computer Science: Normative System Specification*, edited by J.-J. C. Meyer & R. J. Wieringa, 85–100. New York, N.Y.: John Wiley & Sons, 1993.
[158] J. Hintikka. Quantifiers in deontic logic, Societas Scientiarum Fennica, *Commentationes humanarium litterarum* 23, no. 4, 1957.
[159] J. Horty. *Agency and Deontic Logic*, New York, NY: Oxford University Press, 2001.
[160] G. Holmström-Hintikka, S. Lindström, and R. Sliwinski, eds. *Collected Papers of Stig Kanger with Essays on his Life and Work*, Volume I, Synthese Library Vol. 303, Dordrecht: Kluwer, 2001.
[161] S. Kanger. *New Foundations for Ethical Theory, Part 1*, (mimeographed) Stockholm, 1957, reprinted (with minor changes) in R. Hilpinen (ed.): *Deontic Logic: Introductory and Systematic Readings*, Dordrecht: D. Reidel 1971, 36–58. Reprinted in Holmström-Hintikka *et al.*(2001).
[162] J.-J. Ch. Meyer. A different approach to deontic logic: deontic logic viewed as a variant of dynamic logic, *Notre Dame Journal of Formal Logic* 29, 109–136, 1988.
[163] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning.* San Mateo, CA: Morgan Kaufmann Publishers, 1991.
[164] K. Segerberg. Routines. *Synthese* 65, 185–210, 1985.
[165] K. Segerberg. Getting started: beginnings in the logic of action, *Studia logica* 51, 347–378, 1992.
[166] K. Segerberg. Irrevocable belief revision in dynamic doxastic logic, *Notre Dame Journal of Formal Logic* 39, 287–306, 1998.
[167] K. Segerberg. Outline of a logic of action. In *Advances in Modal Logic*, Volume 3, Frank Wolter, Heinrich Wansing, Maarten de Rijke, and Michael Zakharyaschev, editors, 57–84. Lecture Notes. Stanford, CA: CSLI Publications, 2001.
[168] K. Segerberg. Modellings for two types of action. In *A Philosphical Smorgasbord: Essays on Action, Truth and Other Things*, edited by K. Segerberg and R. Sliwinski, 151–156. Uppsala: Filosofiska institutionen. 2003.
[169] K. Segerberg. Trying to meet Ross's challenge. In *Logic and Philosophy in Italy: Essays in honor of Corrado Mangione*, edited by Edoardo Ballo and Miriam Franchella, pp. 155–166. Monza: Italy: Polimetrica, 2006.
[170] G. H. von Wright. Deontic logic, *Mind* 60, 1–15, 1951.
[171] G. H. von Wright. *Norm and Action: A Logical Inquiry.* London: Routledge and Kegan Paul, 1963.

INDEX

0-1 laws, 321

ABox, 46, 766
absorption, 201, 221
actualist quantification, 130, 134, 560
admissible
  rule, 496, 536–541, 795
  concrete domain, 803
AFA axiom, 960
aggregation, 808
AGM postulates, 1024, 1184
$\mathcal{ALC}$, 759
Alexandrov space, see topology
algebra,
  Boolean ∼, 72
    atom in ∼, 348
    atomic ∼, 348
    completion of ∼, 373
    operator in ∼, 340
    compact element, 373
    complete ∼, 384
    complete operator in ∼, 349
    expanded ∼, 340
  Boolean ∼ with operators (BAO), 72, 340
    atom structure of ∼, 349
    complete homomorphism, 350
    opremum in ∼, 346, 506
    radical in ∼, 346
  closure ∼, see topo-Boolean algebra
  complex ∼, 338
  congruence distributive ∼, 342
  directly indecomposable ∼, 345
  formula ∼, 337
  free ∼, 421–422
  Heyting ∼, 476–477, 602
  interior ∼, see topo-Boolean algebra,
  Lindenbaum-Tarski ∼, 341
  simple ∼, 345
  subdirectly irreducible ∼, 345
algebra (*cont.*)
  tense ∼, 384
  topo-Boolean ∼, 951
algebraic semantics of modal logics, 72–75, 331–389
algebraizes, 339
almost perfect information, see information
$\mathcal{ALN}$, 779
alternating automata, see automata
alternating-time temporal epistemic logic, see ATEL
alternating-time temporal logic, see ATL
alternation (in modal μ-calculus)
  depth, 747
  fixpoint, 747
amalgamation property, 369, 525
analytic tableau, see tableau
Anderson's reduction, 997
approximation
  in μ-calculus, 728
  of concept descriptions, 800
arithmetical completeness and interpretation, 934–945
analytically true, 1155
annotation (of states for modal μ-calculus), 732
Aristotelian essentialism, 1156
assignment (in temporal logic)
  local ∼, 663
  non-local ∼, 663
ATEL, 1079, 1142
ATL, 665, 1079, 1138–1142
ATL*, 1139
atom (in Boolean algebra), see algebra
atom structure (of BAO), see algebra
atomic (Boolean algebra), see algebra
autobisimulation, see bisimulation
automata
  alternating
    Büchi ∼, 975, 979

automata (*cont.*)
- alternating (*cont.*)
  - Büchi tree ∼, 976–978, 981
  - parity ∼, 742–744
  - two-way parity ∼, 751
- emptiness problem, 156, 226, 287, 705, 738, 850, 974–978
- nondeterministic
  - Büchi automata, 737, 974, 975, 983
  - Büchi tree automata, 155, 976, 977
- deterministic ∼, 391, 398
- tree ∼, 155–157, 226–227, 704, 738, 1062
- Rabin ∼, 738, 741

axiom of extensionality, 630
axiomatisation problem, 439

back-and-forth, 265, 271, 273
background theory (see also TBox), 46, 185, 200
backjumping, 205, 213
backtracking, 204, 213
backward induction, 1079, 1084, 1087, 1092, 1121, 1122, 1128
BAO, see algebra
Barcan formula, 67, 132–135, 551, 555, 561, 579, 586, 605, 1154, 1168–1172
- converse ∼, 67, 551, 561, 1154, 1168–1172
- in first-order temporal logic, 710

base category, 394
basic
- hybrid language, 49
- modal language, 3
- path logic, 188

basis (of a lattice), 509
- strong ∼, 509

Battle of the Sexes, 1081
BDI logic, 999–1005, 1095
- multi-agent ∼, 1013, 1015

Beck-Chevalley condition, 597
behavioral equivalence, 398
belief
- change, 1024, 1183

belief (*cont.*)
- common ∼, 1012–1014, 1128
- contraction, 1024
- expansion, 1024
- revision, 992, 1009, 1024–1026, 1110, 1127, 1184–1186

Beth definability/property 277, 531, 844
BHK semantics, see Brouwer-Heyting-Kolmogorov semantics
binder, see hybrid logic
bisimulation, 14–17, 263–278
- and bounded morphisms/p-morphisms, 17, 258
- and disjoint unions, 16, 261
- and generated subframes/submodels, 16, 259
- and unraveling, 15, 260
- as a game, 23, 263
- auto∼, 14
- between coalgebras, 398
- characterization
  - of basic modal logic, 21, 282
  - of guarded fragment, 287
  - of hybrid logics, 50, 839
  - of modal μ-calculus, 64, 294, 749
- contraction, 14
- equivalence, 399
- finite ∼, 264
- for game logic, 1134
- for hybrid logic, 50, 838
- guarded ∼, 286
- invariance under ∼
  - for basic modal logic, 18, 258
  - for hybrid logics, 50, 838
  - for modal μ-calculus, 736
- invariant Ptime, 320, 750
- largest ∼, 273
- minimisation, 235
- quantifier, 750
- quotient, 275
- safety, 278
- system, 842
- topo ∼, 77

blocking, 201, 218
- double ∼, 226
- equality ∼, 224
- subset ∼, 224

Boolean algebra, see algebra
Boolean algebra with operators, see algebra
bounded
  fragment, see fragment
  model property (see also finite model property), 146, 707
  morphic image of frames/models, 17, 36, 259
  morphism (p-morphism), see also bisimulation
    between frames, 36, 258
    between models, 17, 19, 258
  variable fragment, see fragment
branch condensing, 205
branching time (see also temporal logics), 662–665, 680–682, 700–701, 712, 980–985
BRCC-8, 954
Brouwer-Heyting-Kolmogorov semantics, 928, 930
Büchi automata, see automata
bull-dozing, 698
bundle (in branching time), 663

C-resolution, 232
call-by-name semantics, 945
call-by-value semantics, 945
canonical
  class, 365
  equation, 365
  extension, 353, 373
  formula (see also $\mathcal{D}$-persistent), 365
  formula for
    logics containing K4, 458
    superintuitionistic logics, 478
    tense logics, 464
  modal logic, 365
  model, 34, 89–93, 308, 697
  model construction, 89–93
Cantor's discontinuum, see topology
category, 594
  cartesian ∼, 594
  lex ∼, 595
causal future modality, see modality 958
characteristic formula, 265
choice function, 732
chronicle, 697
chronological future modality, see modality
Church-Rosser property, 631, 885
Church-Turing thesis, 164
CL, see temporal logics
classical mechanics, 623
classification (of TBoxes), 771
classification problem (for provability logics), 937, 938
clause
  condensed ∼, 192
  definitional ∼, 195
  given ∼, 204
  indecomposable ∼, 196
  modal ∼, 228
  redundant ∼, 192
  split components, 196
  subsumed ∼, 192
  usable ∼, 203
  variant ∼, 188, 191
  worked-off ∼, 203
closure algebra, see algebra
cluster, 698
  depth, 901
coalgebra, 390, 1060
  cofree ∼, 413
  coloring of ∼, 413
  final ∼, 395
  homomorphic image, 403
  homomorphism, 393
  open set, 404
  sub∼, 404
  sum, 405
    conjunct sum of ∼, 406
coalgebraic modal logic, 415
coalition, 1087
  logic, 1078, 1135–1139
cofinal
  subframe logic, 463
  subreduction, 460
cofree coalgebra, see coalgebra
coloring of coalgebra, see coalgebra
combinatory logic, 928
combined logics, 64–66, 869–925
common
  belief, see belief

common (*cont.*)
knowledge, see knowledge
commonsense reasoning, 992, 1023
commutativity axioms, 885
commutator of modal logics, 885
compact (in a Boolean algebra), see algebra
compactness (of a logic)
applications of ∼, 271, 277, 282, 290, 302, 314, 840, 842
invariance of ∼ under simulation, 522
property, 11, 71
non-∼ in conditional logic, 1020
non-∼ in temporal logic, 698
non-∼ of Thomason's consequence, 431
non-∼ of PDL, 60
for concrete domains in description logic, 805
of modal type theory ∼, 648
comparative similarity, 958, 1018–1021
complete information, see information
completion (of Boolean algebra), see algebra
completion tree, 209
complex algebra, see algebra
complexity class, 165
EXPTIME, 165
NEXPTIME, 166
NP, 166
PSPACE, 165
PTIME, 165
computational complexity, 165
of model checking
ATL and ATL*, 1141
CTL, 984
in hybrid logic, 863
LTL, 983
modal formulas, 25
of satisfiability for
ATL, 1142
coalition logic, 1137
conditional logics, 56
CTL, 985
description logics, 776
epistemic logics, 1098, 1101
computational complexity (*cont.*)
of satisfiability of (*cont.*)
fusions of modal logics, 874
game logic, 1133
guarded fragment, 57
hybrid logics, 846–852
logics containing K4.3, 455
logics containing K5, 455
logics containing S4.3, 455
logics containing S5×S5, 158, 456, 890
LTL, 985
metric topological logic, 955
modal μ-calculus, 64, 742
packed fragment, 57
PDL, 61, 726
products of modal logics, 889–910
modal formulas, 30, 198
modal formulas with universal modality, 47
temporal epistemic logics, 1105
tense logics of linear orders, 468
computability logic, 942
computation tree logic, see CTL
concept
description, 759
language, 759
name, 759
pattern, 793
concrete
domain, 801
path, 802
condensing, 192
conditional logic, 54–56, 1017–1021, 1127, 1189–1193
limit assumption, 1018, 1020, 1022, 1189
congruence distributive, see algebra
connected space, see topology
conservativity logic, 940
consequence relation, 494
algebraisable ∼, 511
classical modal ∼, 499
decidable ∼, 496
equivalential ∼, 512
global ∼, 513
global frame ∼, 45

consequence relation (*cont.*)
modal ∼, 499
non-monotonic ∼, 1022–1023
normal modal ∼, 499
Post-complete ∼, 496
preferential ∼, 1022
structurally complete ∼, 496
constant domain model, 67, 131, 552, 580, 884, 1171
constant specification, 943–947
continuity axiom, 606, 607
converse modality, see modality
converse Barcan formula, see Barcan formula
Cook's Theorem, 168
cooperation, 1080
correspondence language, see standard translation
correspondence theory, 232
counterfactuals, see conditional logic
counterpart theory/semantics, 568, 606–616, 1175
covariety, 405
CTL, see temporal logics
CTL*, see temporal logics
Craig interpolation property, see interpolation
Curry-Howard isomorphism, 944
cut rule, 107, 229, 649, 852
analytic ∼, 182, 229
cut-elimination, 107, 229, 632, 956

$\mathcal{D}$-persistent (modal formula), see persistent modal formula
DDL (dynamic deontic logic), see deontic logic
DDL (dynamic doxastic logic), see doxastic logic
de Jongh-Sambin fixed point theorem, see fixed point theorem
de dicto, 561, 569, 575, 1037, 1052, 1107, 1168
de re, 561, 569, 575, 1037, 1052, 1107, 1168
decidable modal logics, see computational complexity
decreasing domain model, 68, 557, 1171
Dedekind complete, see flow of time
deduction theorem, 497
default reasoning, 1023
degree of Kripke incompleteness, see incompleteness
deontic logic, 996–999, 1204–1209,
dynamic ∼, 997–999, 1207–1209
derivational modal logic, 953
derived set, see topology
description logic, 8, 757, 860
descriptive frame, see frame
destructive tableau, see tableau
diagram (of subdirectly irreducible algebra), 506
diamond, 3, 333
compound ∼, 334
induced ∼, 334
difference modality, see modality
difference operator, see modality
direct product, see product
discrete topology, see topology
discriminator
term, 386, 514
unary ∼, 386
variety, 386
disjoint union of frames/models (see also bisimulation), 16, 19, 36, 261
disjunctive property, 542, 944
disjunction property, see disjunctive property
downarrow binder, see hybrid logic
doxastic logic, 993–995,
dynamic ∼, 1026, 1187–1197
multi-agent ∼, 1012–1013
duality theory, 74
Dummett axiom, 459, 698
dynamic
deontic logic, see deontic logic
doxastic logic, see doxastic logic
epistemic logic, see epistemic logic
system, 929, 955–958
topological logic, 955–958
S4C and S4F, 956
topology, 955–956

e-connection, 920
earlier-later relation, 658

effectivity
  α, 1089, 1094, 1135
  β, 1089
  function, 1087–1090, 1094, 1131, 1136
elementary arithmetic, 935
elementary topos, 601
emptiness problem, see automata
Entscheidungsproblem, 444
epistemic
  action, see epistemic program
  logic, 7, 993–995, 1005, 1010–1013, 1022, 1078, 1097–1101
    dynamic ∼, 1078, 1110–1121, 1123, 1124
  program, 1111, 1116
  temporal logic, see temporal epistemic logic
equilibrium
  Nash ∼, 1079, 1081–1083, 1092, 1093, 1096, 1126, 1143
  profile, 1084
  sequential ∼, 1087
  subgame-perfect ∼, 1078, 1084, 1087
essential (element of BAO), see opremum
etale space, 601
evaluation point, 674
evidence function, 947
evidence-based knowledge, 951
existence predicate, 551, 558, 559
existential semantics, 950
existential restriction, 760
expanding domain model, see increasing domain model
explicit reflection, 943
expressive completeness, 54, 679, 684
  Dedekind complete time, 685, 693, 695
  first-order, 693–695
  many-dimensional, 690–691
  non-linear time, 689
  separation and, 687–689
extension, 635, 1153,
  extensional function, 629
  principle of extensionality, 1176

Fagin's theorem, 144
fairness, 1139
feature
  abstract ∼, 802
  concrete ∼, 802
  agreement, 810
  logic, 859
filter, 37, 343, 495
  Leibniz ∼, 513
  modal/open ∼, 344
  principal ∼, 344
  proper ∼, 37, 343
filtration (see also finite model property), 29, 150, 267, 706, 707, 711, 726, 835, 890
  largest ∼, 29, 268
  smallest ∼, 29, 268
  lemma/theorem ∼, 29, 268
final sequence, 397
Fine–van Benthem theorem, 312
finite
  bisimulation, see bisimulation
  character (of set), 90
  depth logic, 454
  variable fragment, see fragment
  width logic, 454
finite model property (fmp), 28, 266
  and finite algebras, 363
  and model generation, 233
  and nominals, 848
  and tense extensions, 473–474
  and the existence of a basis, 509
  for uniform formulas, 449
  invariance under simulation, 522
  no ∼ for $\mathsf{K4} \times \mathsf{K4}$, 902
  of basic modal logic K, 28, 268
  of (cofinal) subframe logics, 473, 475
  of fusions, 874
  of guarded fragment, 57, 287
  of logics containing $\mathsf{S4.3}$, 455, 508
  of logics containing $\mathsf{K5}$, 456
  of logics containing $\mathsf{S5} \times \mathsf{S5}$, 158, 456, 890
  of logics of finite depth, 454
  of CTL and modal μ-calculus, 64, 726, 741
  of PDL and CPDL, 61, 726, 830
  of temporal logics CL/Lin, 706-708
  of two-variable fragment, 279

finite model propery (*cont.*)
  of products of modal logics, 890
  of union-splittings, 436
  undecidability of ∼, 440
first-order
  hybrid logic, 572
  intensional logic, 564
  logic of proofs, 949
  modal logic, 66–68, 130, 549–620, 1152–1175
  S4.3, see S4.3
  S5, see S5
  subframe logics, 588
  temporal logic, 668, 676, 691–695, 704, 710–712
Fitting models, 947, 948
fixed point (in Gödel-Löb logic), 532, 935
  theorem, 532, 935
fixpoint logic, see mu-calculus
flow of time, 658
  cyclical ∼, 661
  examples, 706, 708
  linear ∼, 659
  dense ∼, 659
  Dedekind complete ∼, 54, 659
  continuous ∼, 659
  separable ∼, 660
Floyd–Hoare logic, 723
forcing, 930, 947, 952, 960
forgetful projection, 945, 946, 951
formal epistemology, 928, 933, 941, 947, 950
formula algebra, see algebra
fragment
  of first-order logic
    bounded ∼ 52, 577, 839, 863
    finite/bounded variable ∼, 27, 54, 58, 69, 145, 278–281
    guarded ∼, 56–58, 285–288, 753
    guarded ∼ with constants, 850
    packed ∼, 57
  of first-order modal and temporal logic
    classical ∼, 580
    monadic ∼, 581
    monodic ∼, 582–586, 710–712
    two-variable ∼, 581
frame
  definability, 35
  descriptive ∼, 307, 309, 355
  general ∼, 303, 305, 306, 355, 433
  full general ∼, 303
  incompleteness, see incompletenes
  representation (for first-order modal logic), 587
  total ∼, 664
free algebra, see algebra
functional dependency, 806
functor, 596
  adjoint ∼, 602
  cartesian ∼, 596
  lex ∼, 596
fusion (of modal logics), 65, 872

game
  2-player ∼, 1078, 1084, 1130
  algebra, 1134
  bisimulation ∼, 23
  competitive ∼, 1078, 1081
  concurrent ∼, 1139, 1141
  cooperation in ∼, 1087–1090
  disjunctive ∼, 1133
  dual ∼, 1131
  extensive ∼, 1078, 1080, 1082–1087, 1090–1097, 1126–1130
  finite ∼, 1125
  generic ∼, 1084, 1092
  infinite ∼, 1125
  iterated ∼, 1131
  knowledge, 1099
  outcome of ∼, 1080
  parity ∼, 739
  strategic ∼, 1080–1082, 1089–1091, 1122–1126
  subgame of a ∼, 1084, 1087
  synchronous ∼, 1139
  turn-based synchronous ∼, 1139, 1140
  von Neumann ∼, 1109
  win-loss ∼, 1084
  with simultaneous moves, 1085, 1086
  zero-sum ∼, 1081
game logic, 1078, 1130–1135, 1143
game model, 1131
game theory, 1078–1144

GCI (general concept inclusion axiom), 765
Geach axiom, 459
general frame, see frame
generalized element, 594
generated subframe/substructure (see also bisimulation), 36, 259
generated submodel (see also bisimulation), 16, 19
Gentzen system (see also sequent calculus), 113, 229, 639 702, 946, 1159
geometric morphism, 602
  essential ∼, 605
GF (guarded fragment of first order logic), see fragment
Gödel translation (of intuitionistic logic into S4), 477, 932, 941
Gödel's incompleteness theorem, 931, 933, 936, 938, 942
Gödel-Löb logic, see GL
GL (Gödel-Löb logic), also called G and Löb logic, 929–939
  admissible rules for ∼, 539
  canonical formula for ∼, 462
  completeness of ∼, 91
  expanding products of ∼, 913–917
  fixed point theorem for ∼, 532, 935
  interpolation of ∼, 531
  logics containing ∼, 452, 454
  non-elementarity of ∼, 39, 311
  products of ∼, 909
  provability interpretation of ∼, 929–939
  topological interpretation of ∼, 953–954
  tableau for ∼, 106
  tense extension of ∼, 474
  uniform interpolation of ∼, 535
GL (for game logic), see game logic
global modality, see modality
global satisfiability problem, 32
Goldblatt–Thomason theorem, 41, 312, 360, 840
granularity, 665
Grothendieck topology, 928, 958
Grz, 459, 474, 480, 941
Grzegorczyk axiom, see Grz
guarantee properties, 52
guarded bisimulation, see bisimulation
guarded fragment, see fragment

HA, see Heyting arithmetic
Halldén-completeness, 528
Henkin
  construction, 111, 832
  dimension, 690
Hennessy–Milner
  logic, 723
  property, 271, 299
  theorem, 270
hereditary structural completeness, 542
Heyting arithmetic, 936, 949
highly undecidable problems
  in first-order modal logic, 589
  in hybrid logic, 830, 850
  in product modal logic, 909
  in temporal logic, 701, 710
  Thomason's consequence, 45, 431
Hilbert's classical decision problem, 429
Hintikka set elimination, 151
history (in branching time), see tree
Hoare implication, 955
Horn axiomatisable modal logic, 886
hybrid control systems, 955
hybrid logic, 49–52, 448, 522, 543, 674, 821–864
  binder, 51, 573, 823, 831
  first-order ∼, 572–577
  satisfaction operator, 49, 573, 823
  temporal reference and ∼, 1044
hypercube, 1102
hyperdoctrine,
  classical ∼, 602
  intuitionistic ∼, 602
  modal ∼, 596
hyperintensionality, 632, 649
hyperresolution, see resolution calculus
hypersequents, see sequent calculus

IFP, see mu-calculus
IL, see interpretability logic
IL, see intensional logic
imperfect information, see information
imperfect recall, 1086

implication
  material, 1017
  strict, 1017–1018
incompleteness
  Kripke/frame ∼, 35, 434, 450, 453, 464, 468, 474, 909, 953
  degree of Kripke ∼, 434
  in first-order modal logic, 586–588
  for atomic algebras, 362–365
increasing domain model, 68, 556, 912, 1171
indexicality, 859
indiscernability, 642
inductive formula, 315
inference rule
  admissible ∼, 496, 536-542, 795
  Gabbay-Burgess-style ∼, 829
  irreflexity rule, 543, 701
  modus ponens, 10, 88
  necessitation rule (also called modal generalisation), 10, 88
  of margins, 543
  regularity rule, 88
infinitary
  modal equivalence, 271–273
  modal logic, 20, 69–70, 271–273, 294, 416, 930, 960
infinite combinatorics, 960
inflationary fixpoint logic, see mu-calculus
information
  almost perfect ∼, 1085, 1135, 1138
  complete ∼, 1097, 1098
  imperfect ∼, 1085–1087, 1099, 1107–1110, 1129–1130, 1142
  perfect ∼, 1078, 1082–1085, 1097
information set, 1085, 1087, 1107
information systems, 860
instance checking, 770
Int, see intuitionistic propositional logic
intension, 564, 568, 635, 1036–1037, 1153, 1176
  intensional function, 629
  intensional model, 566, 635
intensional logic (IL), 627, 1178
interior algebra, see algebra
intermediate logic (see also superintuitionistic logic), 588
internalization rule (in logic of proofs), 944
internalization (of TBoxes), 769
interpolation, 21, 276, 369, 498, 525, 695, 750, 844
  uniform ∼, 534, 750, 845
interpretability logic (IL), 939, 940
interpreted system, 1123, 1129–1130
  full system, 1102
  perfect recall in ∼, 1105
  synchronous ∼, 1104
intersection of programs, 828
interval temporal logic, see temporal logics
introspection, 122, 995, 1109
intuitionistic propositional logic, 476, 928–932, 941, 949
intuitionistic logic of proofs, 949
inverse method, 229, 230
inverse modality, see modality
IPC, see intuitionistic propositional logic
irreflexivity rule, see inference rule
isomorphism
  finite ∼, 271
  partial ∼, 271
  potential ∼, 840
  intensional ∼, 1178

joint action, 1139
Jónsson's lemma, 346
Jónsson-Tarski representation theorem, 352
justification logic, 948–951

K, 10
  automated reasoning
    tableau-based, 209–217
    translation-based, 186–200
  completeness of ∼, 87–91
  filtration for ∼, 28, 150
  finite model propery of ∼, 28, 150
  global satisfiability problem for ∼, 32, 151, 200
  interpolation for ∼, 528–529
  natural deduction for ∼, 95–99
  satisfiability problem for ∼, 27, 186, 209
  sequent calculus for ∼, 116

K (*cont.*)
tableau for ∼, 101–105
uniform interpolation for ∼, 535
K4, 33, 87
admissible rules for ∼, 536–540
automated reasoning
tableau-based, 217–219
translation-based, 200–202
axiomatization problem for ∼, 439
completeness of ∼, 91
finite model property of ∼, 473
logics containing ∼, 452, 458–463
natural deduction for ∼, 98
interpolation of ∼, 531
products of ∼, 901
tableau for ∼, 101–105
resolution for ∼, 228
sequent calculus for ∼, 116,
subframe logics containing ∼, 473
K4.1 (see also McKinsey axiom), 41, 462
K4.2, 462
K4.3, 455, 462
K5 (and logics containing K5), 455, 912
K.Alt$_n$, 33, 456, 475
KB (also called B), 87, 89, 184
KT (also called T), 87, 89
KD (also called D, see also SDL), 87, 89
k-variable property, 690
Kamp frame, 664
Kamp's theorem, 54, 685
KARO, 1005, 1006, 1008, 1009
multi-agent, 1010, 1015
Karp's theorem, 272
KM postulates (Katsuno-Mendelzon posulates), 1186
Knaster-Tarski theorem, 62
knowledge (see also epistemic logic)
common ∼, 123, 951, 992, 1011, 1012, 1079, 1100, 1101, 1109, 1113–1116, 1121, 1123, 1124
distributed ∼, 1102
everybody's ∼, 1102
action, see epistemic program
programs, 1108
set, 1110
Kripke completion of a system, 588
Kripke incompleteness, see incompleteness
Kripke model, 4
Kruskal's tree theorem, 456, 468, 915, 957

λ-conversion, 641
labeled deduction, 638, 862
language
context free ∼, 1055
regular ∼, 1058
regular tree ∼, 1062
largest bisimulation, see bisimulation
lattice of normal modal logics, see NExt(L)
lazy unfolding, 201, 221
lcs (least common subsumer), 791
learning, 1116
least
common subsumer, 791
fixed point logic, see LFP
fixpoint logic, see LFP
Leibniz operator, 511
LFP (see also mu-calculus), 42, 316, 321, 751
limit assumption, see conditional logic
limit closure, 700
Lindenbaum construction, 90
Lindenbaum-Tarski algebra, see algebra
linear temporal logic, see temporal logics (LTL),
linguistics, 1031–1073
mathematical ∼, 1054
liveness, 731
local proposition, 1102, 1103
locale, 504
coherent ∼, 508
continuous ∼, 504
spatial ∼, 504
Löb axiom/principle/formula (see also GL), 34, 39, 933
Löwenheim-Skolem
for modal type theory, 649
property, 11, 311, 431
application of ∼, 880
logic of
action, 997–999, 1005–1010, 1197–1203

logic of (*cont.*)
proofs, 929, 932, 933, 942–946, 950
strings, 1055
trees, 1061
logical necessity, see necessity
logical omniscience, 622, 630, 950, 996, 1038, 1098
lossy channel system, 917, 957
LP, see logic of proofs
LPP, 948, 949
LTL, see temporal logics

μ-calculus, see mu-calculus
MacNeille completion, 383
Magari algebra, 940
Makinson's theorem, 34, 435, 507
many-dimensional
connective, 674, 690
evaluation, 666
map (between Boolean algebras)
additive ∼, 340
completely, 349
multiplicative ∼, 340
normal ∼, 340
smooth ∼, 375
master modality, see modality
matching, 795
Matching Pennies, 1081
matrix (logical), 495
maximality principle, 960
McKinsey axiom/formula, 39, 43, 311, 446
McKinsey-Tarski theorem, 956
meaning postulate, 1155
Medvedev logic, 483
mereotopological relations, 954
metaframe, 599
metaphysical necessity, see necessity
metric
modality, see modality
space, 930, 952–955, 957, 958
temporal logics, see temporal logics
Minkowski's spacetime, 661, 958
Minsky machine, 440
modal
μ-calculus, see mu-calculus
companion of superintuitionistic logics, 479
modal (*cont.*)
description logic, 883
predicate logic, see first-order modal logic
reduction principle, 451
theory, 254
type theory, 631
modality
causal future ∼, 958
chronological future ∼, 958
converse ∼, 126, 185, 202, 281
difference, 48, 128
graded ∼, 186, 207, 225, 237, 762
in set theory, 959
inverse ∼, see converse ∼
master ∼, 358, 508, 834
metric ∼, 954, 955
polyadic ∼, 45, 256, 523
transitive ∼, see K4
universal/global ∼, 46, 128, 185, 128, 386, 387, 954, 958
model,
bundle, 606
counterpart ∼ (see also counterpart theory/semantics), 610
existence theorem, 644, 647
functional ∼, 587
hyperdoctrinal ∼, 597
Kripke ∼, 4
named ∼, 829, 836, 837
pointed ∼, 12
presheaf ∼, 604
topological ∼, 604
topos-theoretic ∼, 604
model checking (see also computational complexity), 24–27, 143-145, 982–984
ATL and ATL*, 1141
basic modal logic, 24–26
game logic, 1134
hybrid logics, 862–863
LTL and CTL, 982–984
modal μ-calculus, 735, 739
temporal epistemic logics, 1106
modus ponens, see inference rule
mono, 595
monadic fragment, see fragment

monadic fragment of second-order logic, 432, 669–670, 679, 705–706, 712, 749, 899
  its universal fragment, 39, 232, 310, 686, 705–706
monodic
  formula, 582, 710
  fragment, see fragment
Montagna's principle, 940
Montague's intensional logic, see intensional logic
Morning Star Paradox, 568, 1157
mosaic method, 160, 708
most specific concept, 793
Mostowski operator, 938
msc, see most specific subsumer
MSOL, see monadic second-order logic
mu-calculus (see also LFP)
  alternating ∼, 1142
  modal ∼, 61–64, 293, 721–756, 936, 1097
    with past, 294, 751
    with nominals, 850
  guarded ∼, 294, 753
  inflationary ∼ (IFP), 753
Muddy Children, 124, 1100
multi-dimensional modal logic, 875, 1046

n-provability, 938–939
named model, see model
natural deduction, 95, 702, 854
necessitation rule, see inference rule
necessity
  logical ∼, 1181
  metaphysical ∼, 1181
neighbourhood
  frame, 335, 952–953
  semantics, 75–76, 952–953
  model, 75, 335, 1088, 1131
NExt(L), 434, 501
no forgetting, see perfect recall
no learning, 1106, 1108, 1110, 1111, 1113
nominal, 49, 186, 207, 224, 522, 572, 762, 822
non-deterministic computation, 146
non-logical axiom, 185
non-monotonic logic, 992, 1017, 1022, 1023
non-rigid (designator), see rigid designator
non-standard inferences, 788
nonemptiness problem, see automata
noniterative axiom (see also shallow formula), 451
normal form (formula)
  automaton ∼ (also disjunctive ∼), 744
  conjunctive ∼, 227
  disjunctive ∼, 227
  negation ∼ (NNF), 184
  of degree n, 449
  positive ∼,
    in modal μ-calculus, 729
    in CTL and LTL, 979
normal modal logic, 34
NP complete problem, 168
number restriction, 762

obligation
  conditional ∼, 997, 1016
  contrary-to-duty ∼ (CTD), 997
ontological argument, 622
open
  map, see topology
  neighbourhood, see topology
  set, see topology
outcome function, 1080
OWL-DL, 761

P (conditional logic), 1019, 1023
Π^1_1-hard problems, see highly undecidable problems
p-morphism, see bounded morphism
p-morphic image, see bounded morphic image
PA, see Peano arithmetic
packed fragment, see fragment
partial isomorphism, see isomorphism
path formula, 1000, 1139
PDL, 58–61, 291, 725, 761, 1058, 1130
  combinatory ∼, 829
Peano arithmetic, 934, 936, 937, 939–942, 948, 949, 953
pebble game, 279
Peircean branching time logic, see temporal logics

perfect information, see information
perfect recall, 1086, 1105, 1106, 1108–1109, 1111, 1113, 1119
persistent modal formula, 306–307
  $\mathcal{D}$-$\sim$ (see also canonical), 306–307
polarity, 194
poly-size model property (see also finite model property), 455
polyadic modality, see modality
polynomial functor, 392
  finitary Kripke $\sim$, 392
  Kripke $\sim$, 392
polynomial reduction, 168
possibilist quantification, 130, 134, 560
Post correspondence problem, 957
pre-model, 732
predicate
  abstraction, 135, 564, 569
  modal logic, see first-order modal logic
  temporal logic, see first-order temporal logic
preference relation, 1016, 1022, 1080, 1083
prefixed tableau, see tableau
prefix stability, 190
presheaf, 601
pretabular logic, 454
Prior's connectives F, P, G, H (see also tense logic), 671, 696–698
Prisoner's Dilemma, 1081, 1091
probabilistic belief, 1125
product
  direct $\sim$, 298
  finite model property, 889
  of frames, 875, 877
  of modal logics, 876, 877
    relativised $\sim$, 910, 912
    expanding $\sim$, 912–917
  subdirect $\sim$, 345
  topologies, 953
  ultra$\sim$, 298, 306
proof
  checking, 231
  polynomial, 943–947, 950
  predicate, 931, 942, 945
propositional
  attitudes, 624, 630, 1038, 1176
propositional (*cont.*)
  dynamic logic, see PDL
  quantifiers, 318, 535, 626, 677, 679, 861
protocol, 1103
provability
  algebra, 940, 941
  interpretation, 932–938, 942, 950, 952
  logic (see also GL), 929, 933–938, 940–942, 949, 950
  predicate, 929–938
pseudo-model, 158
public announcement, 1111–1115, 1124
pullback, 402, 594
  weak $\sim$, 402
PTL, see temporal logics

QTC-logic, 890
quantified
  modal logic, see first-order modal logic
  temporal logic, see first-order temporal logic
quasi-variety, see variety
quasimodel, 584, 711, 892, 895, 899

rational behaviour, see rationality
rationality, 1091, 1121–1130
RCC-8, 954
reference point, 674
referential multiplicity, 1163
reflection, 942, 943, 948, 949
region connection calculus, 954
regular
  expressions, 677
  formula, 315
regularity rule, see inference rule
relation lifting, 400
relational domain (in first-order modal logic), 609
relativised product of modal logics, see product
relativistic temporal logic, see temporal logics
resolution calculus, 187
  blocking rule, 201
  closed branch, 193
  complete branch, 193

resolution calculus (*cont.*)
- completeness, 194
- derivation, 193
- expansion rules, 193
- for hybrid logics, 857
- for temporal logics, 704
- hyperresolution, 197
- implementation, 203
- incomplete branch, 193
- inference rules, 193
- modal ∼, 227
- open branch, 193
- open derivation, 193
- ordered chaining, 201
- refutation, 193
- saturation up to redundancy, 193
- selection function, 192
- separation, 206
- splitting, 193, 206
- subsumption, 192

rewriting (of concept descriptions), 799
rigid designator, 135, 551, 555, 564, 668, 1107, 1156, 1164, 1173, 1177, 1183
role
- inverse, 762
- transitive, 762
- hierarchy, 762
- name, 759
- value map, 808

Robinson consistency, 276
root modal, 1038
rule (see inference rule)
Russell-Myhill paradox, 1180

S, 934, 935, 937
Σ^1_1-hard problems, see highly undecidable problems
S4, 34, 87
- admissible rules for ∼, 536–540
- axiomatization problem for ∼, 439
- completeness of ∼, 34, 91
- complexity of ∼, 177
- dynamic topology and ∼, 955–957
- finite model property of ∼, 473
- Gödel's translation of intuitionistic logic into S4, 476–481

S4 (*cont.*)
- interpolation for ∼, 531
- logics containing ∼, 453–455, 526
- natural deduction for ∼, 98
- tableau for ∼, 101–105
- provability interpretation of ∼, 929–941, 946
- topological semantics of ∼, 76–77, 951–953
- sequent calculus for ∼, 116–117

S4C and S4F, see dynamic topological logic
S4.1, 458
S4.2, 458, 586
S4.3
- logics containing ∼, 455, 508, 542
- first-order ∼, 586

S5, 87
- completeness of ∼, 89
- hypersequents for ∼, 117–121
- natural deduction for ∼, 98
- tableau for ∼, 112–113
- first-order ∼, 554, 562, 1152, 1160, 1169

S5 × S5, 158, 456, 882
SAFA axiom, 960
safety, 731
Sahlqvist
- theorem, 40, 315, 366, 382, 446
- formula, 314, 366, 446, 834

SAT, 214, 796
satisfaction operator, see hybrid logic
saturation (in model theory), 270, 299
SDL (see also KD), 996, 997
second-order propositional modal logic, see propositional quantifiers
second-order quantifier elimination, 232
- SCAN, 232
- SQEMA, 232

Segerberg's axiom, 61
selection of points, 149
self-reference, 928, 946, 947
semantic tableau, see tableau
separation property, 687–689
sequent calculus, 113, 229
- analytic cut rule, see cut rule
- backward reasoning, 229

sequent calculus (*cont.*)
  cut rule, see cut rule
  for hybrid logic, 852
  for temporal logic, 702
  forward reasoning, 229
  hypersequents, 117
  logical rules, 229
  proof, 229
  structural rules, 229
set theory, 959
  non-well-founded ∼, 69, 959
shallow formula (see also noniterative axiom), 316
sheaf theory, 601–606, 958
$\mathcal{SHIQ}$, 761
$\mathcal{SHOIQ}$, 761
signed formula, 99
simple algebra, see algebra
simulation (between modal logics), 517
social choice theory, 1080, 1087, 1144
solution concept, 1078, 1081, 1092–1094, 1121–1130
Solovay's Theorem, 934
spatial reasoning, 76, 860, 951
special theory of relativity, 958
sphere system, 1189
splitting, 435, 505
  companion, 505
  representation, 508
  subframe ∼, 475
  theorem, 506
standard deontic logic, see SDL and KD
standard translation, see translation
state formula, 682, 999, 1139
Stavi connectives, 673, 686, 694
stit-theory, 1198
Stone duality, 355
Stone space, see topology
strategy, 1080, 1103, 1139
  best response ∼, 1093, 1096, 1123
  dominant ∼, 1082
  dominated ∼, 1082, 1122, 1124, 1127
  joint ∼, 1138
  mixed ∼, 1082
  profile, 1080, 1092
  pure ∼, 1082
  uniform ∼, 1085, 1107
strategy (*cont.*)
  winning ∼, 1084
strong provability, 941
structural theory of sets, see STS
structural
  transformation, 194
  pre-completeness, 542
  subsumption, 778
STS, 930, 960
sub-Boolean description logics, 773
subdirect product, see product
subdirectly irreducible, see algebra
subframe logics, 463, 469–475, 531, 911
  first-order ∼, 588
subobject, 595
subordinate proof, 95
subreduction, 459
subsumption (between concepts), 767
succinctness, 686
superamalgamation property, 370, 525
superintuitionistic logics, 476–483, 527
synonymy, 568
syntax
  model theoretic, 1070
  natural language, 1054–1072

tableau calculus, 99–113, 208, 528
  analytic ∼, 107
  and automated reasoning, 207–226
  and interpolation, 528, 535
  complete ∼, 210
  destructive ∼, 101
  tableau expansion rule, 208
  for hybrid logic, 855
  for modal type theory, 638
  for temporal logics, 702–704
  labelled ∼, 208
  open ∼ 210
  prefixed ∼, 108
  semantic ∼, 99
tabular logic, 453
TBox, 31, 185, 763
  acyclic, 763
  with descriptive semantics, 763
  with fixpoint semantics, 764
temporal epistemic logic, 667, 882, 1103–1106

temporal logics
branching time
CTL, 682, 700, 704, 723, 735, 831, 980–985, 1079, 1139
CTL*, 292, 681, 689, 700, 725, 831, 1139
first-order CTL and CTL*, 712
PCTL* (CTL* with past), 681, 700
Peircean branching time logics, 681
interval temporal logics, 675–676
linear time
ETL($\mathcal{T}$), 678
first-order LTL, 712
LTL (also called PTL and PLTL), 673, 699, 979–980, 983, 985, 1057, 1105
logics containing Lin/CL, 464–469, 706–707
TLA, 677
UYF and USF, 678–679
metric temporal logics, 674
relativistic temporal logic, 661, 958
temporal
structure, 659
reasoning, 696–711, 971–985
tense
algebra, see algebra
logic, 6, 384, 463, 696
logic of ∼, 1041, 1044
natural language, 1040
reference time, 1043
terminal object, 594
Terminological Box, see TBox
terminology, see TBox
theory
resolution, 203
unification, 203
thrashing, 212
three variable fragment, see fragment
tiling problem
corridor ∼, 170
games, 171
recurrent ∼, 432, 590, 710
square ∼, 168
$\omega \times \omega$ ∼, 44, 147, 581, 905, 910
topo-Boolean algebra, see algebra
topological
duality, 354
model/semantics 76–79, 951–958
space, 77, 951, 1189
topology,
Alexandrov space, 956, 957
Cantor's discontinuum, 953
closed set, 1189
clopen set, 354, 1189
connected space, 79
continuous map, 78, 955
derived set, 930, 953–954
discrete ∼, 77
homeomorphism, 956–957
open
map, 78
neighbourhood, 77
set, 77, 1189
scattered ∼, 953, 954
Stone space, 355, 1190
trivial ∼, 77
topos theory, 601–606, 958
transaction time, 666,
transition system, 665, 730, 972
translation
axiomatic ∼, 201
from many modal operators to one, 516
from polyadic modal operators to unary, 522
functional ∼, 236
optimised functional ∼, 188, 194, 199
polyadic optimised functional ∼, 190
relational ∼, 186, 199
semi-functional ∼, 236
standard ∼
of first-order modal logic, 563
of first-order temporal logic, 691
of hybrid logics, 832, 838
of modal logic, 10–11, 254
of modal logic into second-order logic, 38, 318, 432
of temporal logic, 683
tree-based relational ∼, 199
tree (in branching time), 660
branch of ∼, 660
history, 662

tree (*cont.*)
  path, 662
tree automata, see automata
tree model property, 261, 295
  finite ∼ of basic modal logic (K), 30, 150, 269
  of modal μ-calculs, 736
  generalized ∼ of guarded fragment, 287
tree unraveling, see unraveling
truth value, 495
  designated, 495
two variable fragment, see fragment

ultrafilter, 37, 351
  principal ∼, 38
  extension, 37, 297, 841
  frame, 304, 352
  morphic image, 841
ultraproduct, see product
unfolding/unraveling, 15, 260
unification, 537, 794
uniform formula, 449
unique name assumption, 766
universal
  modality, see modality
  monadic second-order logic, see monadic second-order logic
unraveling, see unfolding/unraveling
until and since, see temporal logics
update, 1111, 1114–1121, 1186
urelement, 959
utility, 1080
  expected ∼, 1128
  function, 1080, 1083, 1126
  maximal expected ∼, 1126

valid time, 666
value restriction, 760
van Benthem formula, 313
van Benthem theorem, 21, 282
van Benthem–Rosen theorem, 284
variety (of algebras), 420
  complete ∼, 363
  discriminator ∼, see discriminator
  quasi-∼, 511
  semisimple ∼, 514
varying domain model, 67, 134, 557, 1171
Vietoris construction, 392

ω-transitive, see weak transitivity
weak transitivity, 358, 508
Worm Principle, 942

XPath, 689

yesterday connective, 672
ZF, 930, 940, 959
ZFC, 523, 960